



UNIVERSITÀ DI PISA

FACOLTÀ DI INGEGNERIA

Tesi di Laurea in Ingegneria Aerospaziale

**SVILUPPO DI UNA SONDA MULTI-PRESSIONE MEDIANTE ANALISI CFD ED
ALGORITMI BASATI SU RETI NEURALI PER LA RICOSTRUZIONE DEI
PARAMETRI DI VOLO**

Relatori:

Prof. Ing. **Roberto Galatolo**

Prof. Ing. **Eugenio Denti**

Ing. **Francesco Schettini**

Ing. **Veronica Poggi**

Ing. **Alberto Calia**

Candidato:

Marco Solano

Anno Accademico 2007-2008

*Per mio padre
e mia madre*

Sommario

Il presente lavoro riguarda lo studio di fattibilità di una sonda multi-pessione per la determinazione dei parametri di volo, quali gli angoli di incidenza e derapata, la quota di volo ed il numero di Mach. La determinazione di tali parametri riveste un ruolo di particolare importanza nella riuscita della missione di velivoli da trasporto civile, militare e velivoli di piccole dimensioni non abitati.

La sonda (oggetto di studio) progettata si ispira ai sistemi di tipo *Flush Air Data Sensor System* (FADSs), ovvero sistemi di sensori dati aria di tipo non intrusivo composti da una serie di prese di pressione generalmente realizzate nella parte prodiera della fusoliera, le cui misure vengono elaborate da algoritmi di calcolo appositamente dedicati.

Il lavoro è stato diviso in due parti principali. Nella prima parte è stata definita la geometria della sonda sulla base di alcune architetture presenti in commercio ed è stata effettuata un'analisi di fluidodinamica computazionale (*Computational Fluid Dynamics*, CFD) sulla sonda in configurazione “*stand alone*” che ha portato alla definizione di un database aerodinamico preliminare. Tale database ha permesso di conoscere le pressioni misurate nelle apposite prese al variare dell'assetto e della velocità.

Le misure di pressione fornite dal database sono state utilizzate in fase di addestramento delle reti neurali, descritte nella seconda parte del lavoro. L'uso di architetture neurali è un approccio innovativo alla ricostruzione dei parametri di volo, alternativo alle tecniche di tipo classico quali, ad esempio, quelle basate su funzioni polinomiali di taratura. Nella fase di calibrazione delle reti neurali si sono tenute in considerazione solamente le reti dedicate al calcolo degli angoli di assetto e della velocità di volo in quanto è pensabile risalire alla pressione statica aggiungendo sulla sonda un foro di misura disposto tangenzialmente al flusso con funzionamento analogo ad un Tubo di Pitot. Durante la fase di addestramento delle

reti è stato effettuato uno studio di ottimizzazione sia sul numero di prese di pressione sia sulla struttura della rete che sulla tipologia di addestramento.

Indice

SOMMARIO	I
INDICE	III
INDICE DELLE FIGURE	V
INDICE DELLE TABELLE.....	VIII
INTRODUZIONE	IX
PARTE I: DEFINIZIONE DELLA GEOMETRIA DI UNA SONDA MULTI-PRESSIONE E GENERAZIONE DI UN DATABASE PRELIMINARE	3
1 DESCRIZIONE DEL SISTEMA DATI ARIA E DELLE VARIE TIPOLOGIE DI SENSORI.....	1
1.1 IL SISTEMA DATI ARIA	1
1.2 TIPOLOGIE DI SENSORI DATI ARIA.....	1
1.3 DEFINIZIONE DELLA GEOMETRIA DELLA SONDA MULTI-PRESSIONE.....	4
2 GENERAZIONE DELLA MESH.....	10
2.1 INTRODUZIONE.....	10
2.2 DESCRIZIONE DEL SOFTWARE <i>GAMBIT</i> [®]	11
2.3 REALIZZAZIONE DELLA MESH	22
2.4 CONDIZIONI AL CONTORNO	30
3 GENERAZIONE DEL DATABASE AERODINAMICO MEDIANTE ANALISI CFD.....	32
3.1 INTRODUZIONE A FLUENT [®]	32
3.2 IMPOSTAZIONE DELLE SIMULAZIONI.....	35
3.3 POSIZIONAMENTO DEI PUNTI DI CONTROLLO	42
3.4 ANALISI DEI RISULTATI DELLE SIMULAZIONI CFD	44
PARTE II: RICOSTRUZIONE DEI PARAMETRI DI VOLO MEDIANTE RETI NEURALI	58
4 LE RETI NEURALI	59
4.1 CAMPI D'APPLICAZIONE E POSSIBILI PROBLEMATICHE	59
4.2 RETI NEURALI STATICHE E METODOLOGIE DI ADDESTRAMENTO.....	63
4.2.1 <i>Metodo di addestramento Early Stopping</i>	67
4.2.2 <i>Metodo di addestramento Bayesian Regularization</i>	68
5 INDIVIDUAZIONE DEI PARAMETRI PRINCIPALI DELLE RETI.....	70
6 RISULTATI E VALUTAZIONE DELLE PRESTAZIONI DELLE RETI.	81
6.1 PARAMETRI DI CONFRONTO DELLE RETI NEURALI	83

6.2	RETI A SINGOLA USCITA	86
6.3	RETI A DOPPIA USCITA.....	97
6.4	RISULTATI GENERALI	101
7	CONCLUSIONI E SVILUPPI FUTURI	104
	BIBLIOGRAFIA.....	105
	APPENDICE A-ERRORI COMMESSI SUL TRAINING SET	107
	APPENDICE B-ERRORI COMMESSI SUL TESTING SET	114

Indice delle figure

<i>Fig. 1.1 Tubo di Pitot</i>	2
<i>Fig. 1.2 Sonda Multi-probes</i>	2
<i>Fig. 1.3 Nose Shuttle X33</i>	3
<i>Fig. 1.4 Disposizione prese di pressione Shuttle X33</i>	4
<i>Fig. 1.5 Geometria preliminare della sonda</i>	5
<i>Fig. 1.6 Vista tridimensionale della sonda</i>	6
<i>Fig. 1.7 Vista della piastra</i>	7
<i>Fig. 1.8 Vista supporto verticale</i>	8
<i>Fig. 1.9 Sezione supporto verticale</i>	8
<i>Fig. 1.10 Giunzione tra i supporti</i>	9
<i>Fig. 1.11 Vista tridimensionale della sonda con supporto</i>	9
<i>Fig. 2.1 Modello semplificato della sonda</i>	10
<i>Fig. 2.2 Flusso logico dell'analisi CFD</i>	11
<i>Fig. 2.3 Visualizzazione del box esterno</i>	13
<i>Fig. 2.4 Visualizzazione dei box più vicini al modello</i>	14
<i>Fig. 2.5 Visualizzazione dei box più esterni</i>	14
<i>Fig. 2.6 Visualizzazione tridimensionale dei box</i>	15
<i>Fig. 2.7 Menu "Mesh Faces"</i>	16
<i>Fig. 2.8 Applicabilità degli elementi di superficie al modello di riempimento</i>	17
<i>Fig. 2.9 Raffigurazione dei modelli di riempimento della mesh</i>	17
<i>Fig. 2.10 Inserimento della spaziatura tra i nodi</i>	19
<i>Fig. 2.11 Visualizzazione tridimensionale degli elementi di volume</i>	20
<i>Fig. 2.12 Menu "Mesh Volumes"</i>	20
<i>Fig. 2.13 Scelta del modello di riempimento</i>	21
<i>Fig. 2.14 Applicabilità degli elementi di volume al modello di riempimento</i>	22
<i>Fig. 2.15 Mesh di superficie della parte anteriore della sonda</i>	23
<i>Fig. 2.16 Mesh superficiale del modello</i>	24
<i>Fig. 2.17 Visualizzazione dei nodi lungo una linea di contorno tra due superfici</i>	25
<i>Fig. 2.18 Visualizzazione della mesh superficiale sul primo box</i>	26
<i>Fig. 2.19 Visualizzazione della mesh superficiale completa del primo box</i>	26
<i>Fig. 2.20 Visualizzazione della mesh superficiale sul secondo box</i>	27
<i>Fig. 2.21 Visualizzazione della mesh superficiale completa del secondo box</i>	27
<i>Fig. 2.22 Visualizzazione della mesh superficiale del modello completo</i>	28
<i>Fig. 2.23 Elementi lungo le superfici di contorno tra volumi</i>	29
<i>Fig. 2.24 Sezione della mesh di volume del modello</i>	30
<i>Fig. 2.25 Visualizzazione menu d'inserimento delle condizioni al contorno</i>	31
<i>Fig. 3.1 Flusso logico dell'impostazione dei parametri della simulazione</i>	35
<i>Fig. 3.2 Visualizzazione "Grid Check"</i>	36
<i>Fig. 3.3 Visualizzazione Menu "Solver"</i>	37
<i>Fig. 3.4 Visualizzazione Menu "Viscous Model"</i>	38
<i>Fig. 3.5 Visualizzazione Menu "Materials"</i>	39
<i>Fig. 3.6 Visualizzazione Menu "Velocity Inlet"</i>	40

Fig. 3.7 Visualizzazione del generico volume di controllo	41
Fig. 3.8 Visualizzazione Menu "Solution Controls"	41
Fig. 3.9 Visualizzazione degli archi sulla sonda	42
Fig. 3.10 Visualizzazione della spaziatura sugli archi principali	43
Fig. 3.11 Visualizzazione punti di controllo	44
Fig. 3.12 Flusso logico per la convergenza della soluzione	45
Fig. 3.13 Andamento dei residui	46
Fig. 3.14 Andamento della velocità sul piano xy	47
Fig. 3.15 Visualizzazione della pressione statica sul piano xy	47
Fig. 3.16 Visualizzazione della pressione dinamica sul piano xy	48
Fig. 3.17 Andamento di ΔP al variare di α con $\beta=0^\circ$ per un flusso a 20 m/s	49
Fig. 3.18 Andamento di ΔP al variare di α con $\beta=0^\circ$ per un flusso a 20 m/s	49
Fig. 3.19 Andamento di ΔP al variare della velocità fissati $\alpha=0^\circ$ e $\beta=0^\circ$	50
Fig. 3.20 Andamento di ΔP al variare della velocità fissati $\alpha=10^\circ$ e $\beta=0^\circ$	51
Fig. 3.21 Visualizzazione simmetrie geometriche	51
Fig. 3.22 Griglia simulazioni a 20m/s in funzione di α e β	52
Fig. 3.23 Griglia simulazioni a 70m/s in funzione di α e β	52
Fig. 3.24 Andamento ΔP al variare di α e β per	53
Fig. 3.25 Grafico errore commesso durante le interpolazioni	56
Fig. 3.26 Andamento ΔP al variare di α e β per un flusso a 20 m/s	57
Fig. 4.1 Schema di un neurone biologico	60
Fig. 4.2 Neurone artificiale: schema di funzionamento	61
Fig. 4.3 Multi-Layer Perceptron ad un solo strato nascosto	64
Fig. 4.4 Algoritmo di Newton	66
Fig. 5.1 Modello Simulink per l'ampliamento del database preliminare	71
Fig. 5.2 Limitazione dell'ingresso tramite blocco di saturazione	71
Fig. 5.3 Lookup table	72
Fig. 5.4 Menù d'impostazione della lookup table	72
Fig. 5.5 Rappresentazione punti di bordo	73
Fig. 5.6 Rappresentazione del Training set, Validation set e Testing set	74
Fig. 5.7 Rappresentazione dei bin	74
Fig. 5.8 Punti scelti come sensori per la sonda	76
Fig. 5.9 Andamento delle pressioni al variare di α nei punti scelti per posizionare i sensori di misura	76
Fig. 5.10 Andamento delle pressioni al variare di β nei punti scelti per posizionare i sensori di misura	77
Fig. 5.11 Funzione heaviside	78
Fig. 5.12 Funzione lineare	78
Fig. 5.13 Funzione lineare a tratti	79
Fig. 5.14 Funzione sigmoide logaritmica	80
Fig. 5.15 Funzione iperbolica	80
Fig. 6.1 Denominazione delle reti	83
Fig. 6.2 Schema di calcolo dell'errore commesso dalle reti	84
Fig. 6.3 Distribuzione degli errori nella stima di α in funzione di α , β e velocità	91

Fig. 6.4 Distribuzione degli errori nella stima di β in funzione di α , β e velocità	94
Fig. 6.5 Distribuzione degli errori nella stima della velocità in funzione di α , β e velocità	95
Fig. 6.6 Distribuzione degli errori nella stima di α in funzione di α , β e velocità per le reti a doppia uscita	100
Fig. 6.7 Distribuzione degli errori nella stima di β in funzione di α , β e velocità per le reti a doppia uscita	100
Fig. 6.8 Andamento MSE durante addestramento rete Net_NSI_0_5_19_33_47_Out_Vel_es	102
Fig. 6.9 Andamento SSE durante addestramento rete Net_NSI_0_5_19_33_47_Out_Vel_br	103
Fig. A.1 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoA_br sul training set	107
Fig. A.2 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoA_br sul training set	108
Fig. A.3 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoA_br sul training set	108
Fig. A.4 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoA_br sul training set	109
Fig. A.5 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoS_br sul training set	109
Fig. A.6 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoS_br sul training set	110
Fig. A.7 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoS_br sul training set	110
Fig. A.8 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoS_br sul training set	111
Fig. A.9 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoA_AoS_br sul training set	111
Fig. A.10 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoA_AoS_br sul training set	112
Fig. A.11 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoA_AoS_br sul training set	112
Fig. A.12 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoA_AoS_br sul training set	113
Fig.B.1 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoA_br sul testing set	114
Fig.B.2 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoA_br sul testing set	115
Fig.B.3 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoA_br sul testing set	115
Fig.B.4 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoA_br sul testing set	116

<i>Fig.B.5 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoS_br sul testing set.....</i>	116
<i>Fig.B.6 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoS_br sul testing set.....</i>	117
<i>Fig.B.7 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoS_br sul testing set.....</i>	117
<i>Fig.B.8 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoS_br sul testing set.....</i>	118

Indice delle Tabelle

<i>Tab. 3.1 Tabella riassuntiva del database preliminare</i>	54
<i>Tab. 6.1 Prima tipologia di reti</i>	81
<i>Tab. 6.2 Seconda tipologia di reti.....</i>	82
<i>Tab. 6.3 Tabella degli errori commessi dalle reti a singolo output α.....</i>	87
<i>Tab. 6.4 Tabella degli errori commessi dalle reti a singolo output β.....</i>	88
<i>Tab. 6.5 Tabella degli errori commessi dalle reti a singolo output Velocità</i>	89
<i>Tab. 6.6 Eccedenze ed errore percentuale relativo alle reti a singola uscita in α</i>	92
<i>Tab. 6.7 Errori reti ad singola uscita (Vel) riaddestrate con training set ampliato</i>	93
<i>Tab. 6.8 Eccedenze ed errore percentuale delle reti a singola uscita in β.....</i>	95
<i>Tab. 6.9 Eccedenze ed errore percentuale delle reti a singola uscita nella velocità</i>	96
<i>Tab. 6.10 Tabella errori commessi dalle reti a doppia uscita</i>	98

Introduzione

Il sistema dati aria di un moderno velivolo di tipo *Fly by Wire* (FBW) ha il compito di determinare i parametri di volo (pressione statica, velocità, angoli di incidenza α e derapata β) a partire dalla conoscenza di grandezze di flusso locale misurate da apposite sonde installate generalmente sulla fusoliera dei velivoli. La derivazione dei parametri di volo, a partire dalle misurazioni delle sonde, viene di solito affidata ad un'unità di calcolo dedicata o ai computer di bordo.

Sulla base delle esperienze maturate presso il Dipartimento di Ingegneria Aerospaziale dell'Università di Pisa (DIA), finalizzate alla definizione di metodi e modelli per l'integrazione dei dati aria nel *Flight Control System* (FCS) di velivoli *Fly-by-Wire*, si è pensato di sviluppare una sonda di architettura non convenzionale ispirata ai *Flush Air Data Sensor System* (FADSs).

I FADSs sono sistemi di tipo non intrusivo poiché costituiti da un insieme di prese di pressione realizzate direttamente sulla parte prodiera della fusoliera, le cui misure sono elaborate da opportuni algoritmi, per la ricostruzione dei parametri di volo. Il numero elevato di prese di pressione che caratterizza tali sistemi consente il funzionamento anche in caso di perdita di una o più misure per il verificarsi di una o più avarie. Quest'aspetto rende le FADSs particolarmente adatte a sistemi di tipo FBW caratterizzati da architetture a molteplice ridondanza.

Nel presente lavoro è stata ideata una sonda multi-pressione da applicare in particolare su velivoli di tipo *Unmanned Aerial Vehicle* (UAV), e sono stati sviluppati opportuni algoritmi per la ricostruzione dei parametri di volo basati su Reti Neurali (*Neural Networks*, NN). Per la scelta della geometria definitiva sono state esaminate varie forme già presenti in letteratura, tutte sottoposte a studi fluidodinamici CFD, realizzati con il *software Fluent*[®]. Fissata la geometria di riferimento si è generato un *database* preliminare contenente le pressioni rilevate sulla sonda in funzione delle condizioni di volo esaminate.

Tale *database* è stato utilizzato per la calibrazione delle NN: durante questa fase del lavoro è stato possibile determinare il numero minimo di prese di pressione necessarie a garantire l'opportuna accuratezza sui parametri ricostruiti.

Nell'ultima parte del lavoro è stato fatto uno studio di ottimizzazione dell'architettura delle reti neurali, variando i parametri principali quali numero di strati e neuroni, funzioni di trasferimento ed algoritmo di addestramento. Tutti i codici di calcolo sviluppati in quest'ultima parte del lavoro sono stati realizzati in ambiente *Matlab*[®].

PARTE I: *Definizione della geometria di una sonda
multi-pressione e generazione di un database
preliminare*

1 Descrizione del sistema dati aria e delle varie tipologie di sensori

1.1 Il sistema dati aria

I moderni velivoli civili e militari sono dotati di sistemi di controllo del volo capaci di assicurare buone caratteristiche di stabilità e manovrabilità. Tali sistemi sono in grado di elaborare i dati forniti dall'*Air Data System* (ADS) ed assistere il pilota nel controllo della traiettoria o controllare automaticamente la risposta del velivolo alle perturbazioni. Un ADS è un insieme di dispositivi che hanno il compito di stimare la velocità di volo, la pressione statica e gli angoli d'incidenza e derapata a partire da misure di pressione del flusso locale acquisite da più sonde.

L'uso da parte dei moderni FCS di tali informazioni impone a questi sistemi d'avere architetture a molteplice ridondanza ed in grado di garantire una certa sicurezza ed affidabilità nonché di fornire un'adeguata precisione nel calcolo di tali informazioni.

1.2 Tipologie di sensori dati aria

In letteratura sono presenti vari tipi di sensori con forme e funzionamento spesso molto differenti: per definire la geometria della nostra sonda quindi sono state valutate diverse tipologie cercando di integrarne le caratteristiche principali.

Una delle apparecchiature più semplici nonché più utilizzate è il tubo di Pitot. Questa sonda è composta da due prese di pressione (vedi *Fig. 1.1*), una per la pressione totale, sistemata sull'estremità anteriore e disposta frontalmente al flusso ed una per la pressione statica sul tubo, disposta tangenzialmente alla corrente. Il Tubo di Pitot viene utilizzato per il calcolo della velocità asintotica del flusso a

partire dalle misure di pressione: prendendo spunto dalle dimensioni di questa sonda si è valutato quelle che potevano essere le dimensioni della nostra sonda.

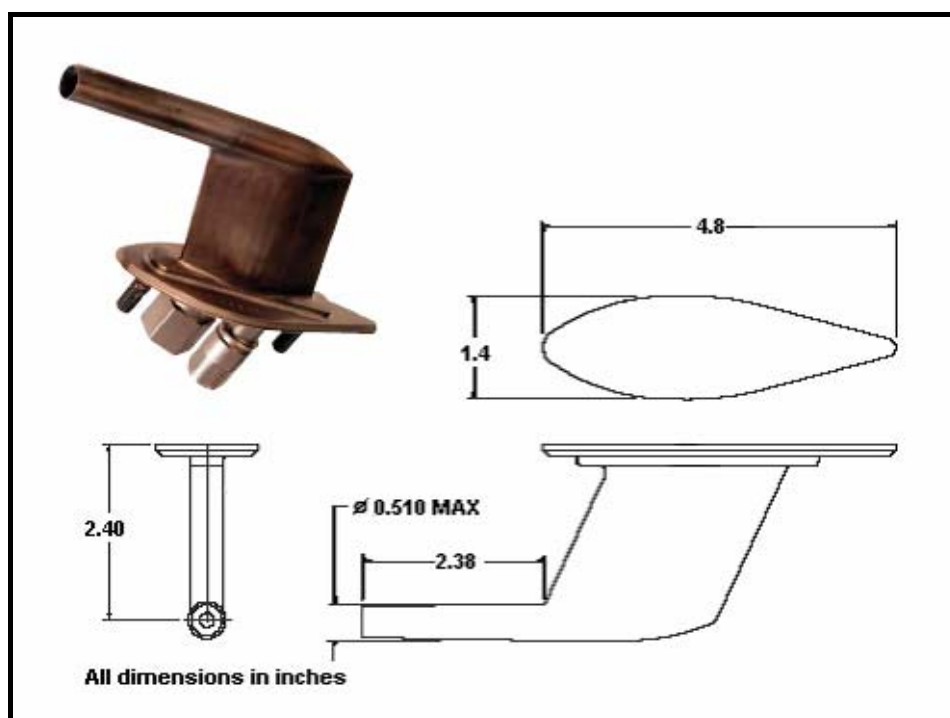


Fig. 1.1 Tubo di Pitot

Inoltre si è preso in considerazione anche sonde di tipo multi funzione presenti in commercio ed utilizzate per il calcolo degli angoli di incidenza e derapata come quella rappresentata in Fig. 1.2.

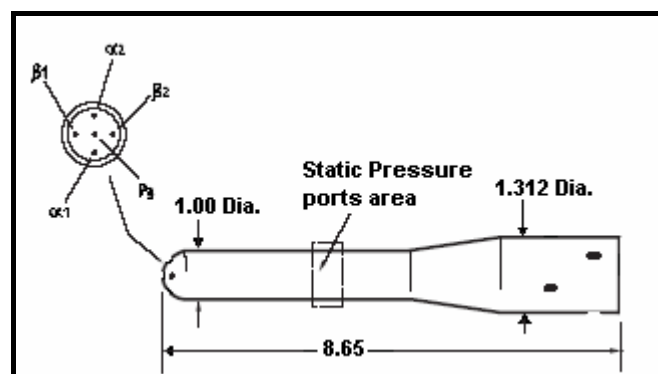


Fig. 1.2 Sonda Multi-probes

Dallo studio della geometria di queste apparecchiature, si è valutata la possibilità di utilizzare un'estremità con superficie sferica sulla quale posizionare le prese di pressione.

I FADSs presentano un numero elevato di fori (da 5 a 27) che forniscono misure di pressione locale: tali sistemi garantiscono un opportuno livello di ridondanza ovvero lo strumento fornisce misure attendibili anche in presenza di eventuali avarie o eventuali condizioni d'ombra.

Questi sistemi sono impiegati negli *shuttle* e in velivoli sperimentali ad elevate prestazioni come l'F18 HARV (*High Alpha Research Vehicle*) [2]: tutte le prese di pressione dei sistemi FADSs sono disposte su di una calotta sferica che di solito è la superficie stessa del *nose* del velivolo ed in particolare sono disposte come in *Fig. 1.3* [4]. Le misure di pressione locale riscontrate nei vari punti di controllo vengono inviate ad un'unità di calcolo dedicata che ricostruisce gli angoli di incidenza e derapata.

Le *Fig. 1.3* e *Fig. 1.4* mostra la disposizione delle prese di pressione identificata da un sistema di coordinate.

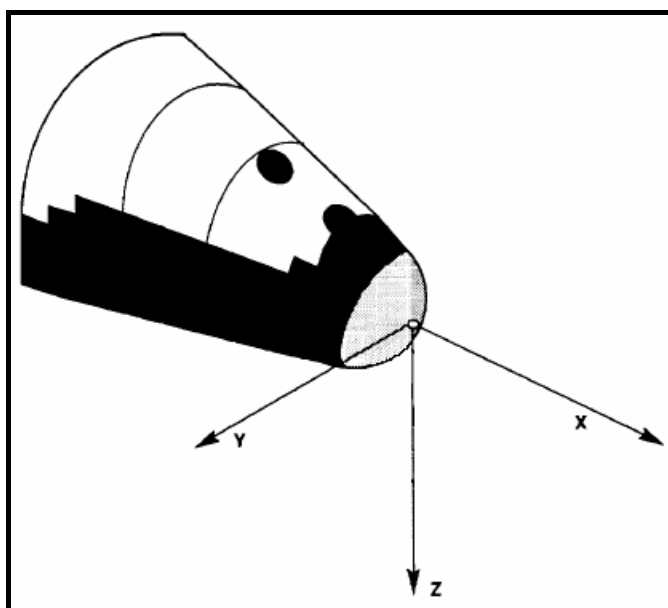


Fig. 1.3 Nose Shuttle X33

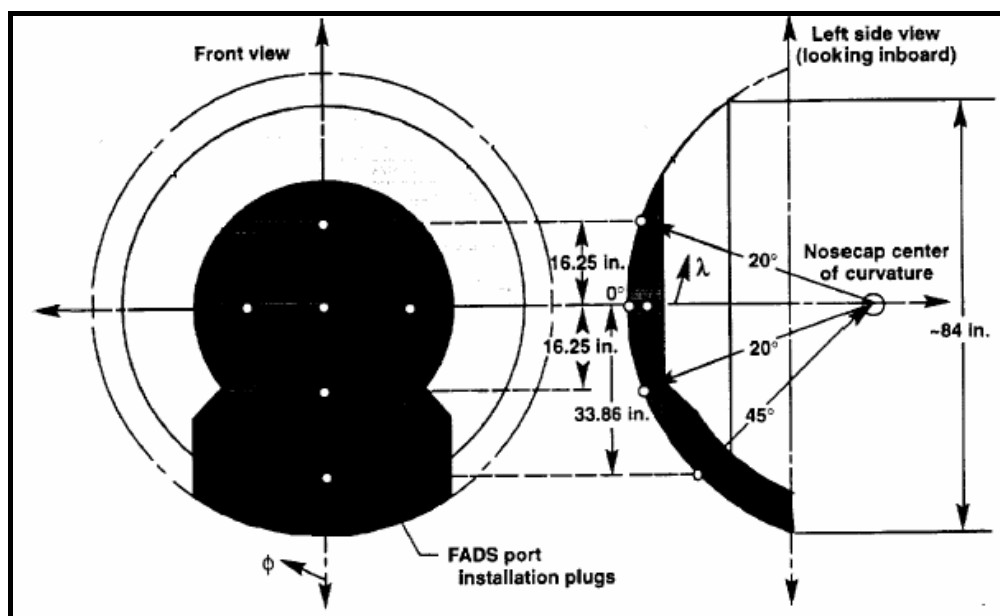


Fig. 1.4 Disposizione prese di pressione Shuttle X33

Sulla sonda sviluppata si è cercato di mantenere le caratteristiche principali dell'architettura di un sistema FADSs, riducendone gli ingombri in modo tale da realizzare una sonda che è possibile installare anche su velivoli non abitati quindi di piccole dimensioni. Tali velivoli spesso presentano sul muso della fusoliera apparecchiature di monitoraggio del territorio: in questi casi è quindi indispensabile l'utilizzo di una sonda di piccole dimensioni ed indipendente, installata tramite un supporto sotto la fusoliera.

1.3 Definizione della geometria della sonda multi-pressione

Presi in considerazione tutti gli elementi precedentemente descritti, le caratteristiche principali che deve avere la nostra sonda possono essere sintetizzate di seguito:

- La sonda deve avere un'architettura simile ai moderni FADSs ma deve presentare un supporto per l'installazione su velivoli UAV, come le normali strumentazioni presenti in commercio.
- Le dimensioni devono essere ridotte il più possibile essendo una sonda di tipo intrusivo.
- Bisogna fare in modo che la geometria della sonda non crei problemi di natura aerodinamica, elevata resistenza o punti di distacco del flusso.

Partendo da tali presupposti, la sonda è stata disegnata con il *software* di modellazione tridimensionale *Catia*[®] (versione 5r12), generando la geometria mostrata in *Fig. 1.6* [10].

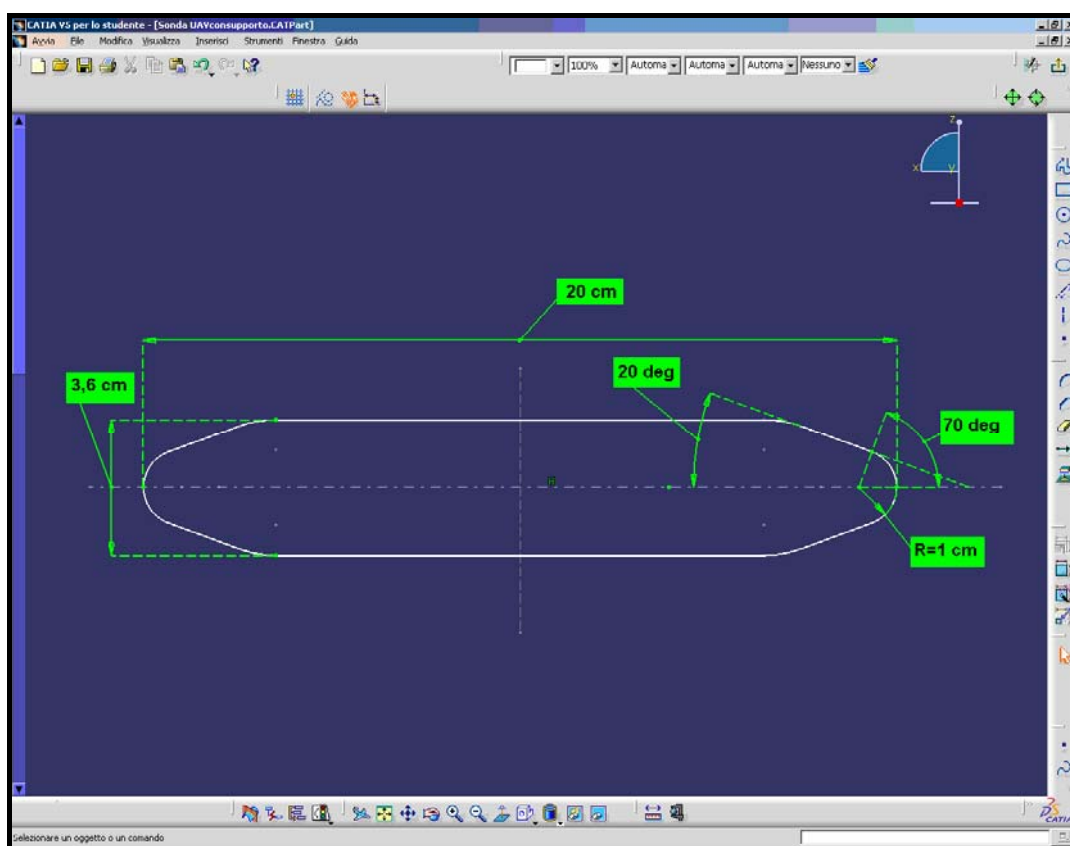


Fig. 1.5 Geometria preliminare della sonda

La parte anteriore della sonda è costituita da una calotta sferica di raggio 1 cm per l'alloggiamento delle prese di pressione e da un corpo cilindrico di diametro 3.57 cm, raccordato alla calotta attraverso una superficie tronco conica di semiapertura 20 gradi. Tale raccordo permette di non avere interferenze aerodinamiche ad alte incidenze e garantisce una superficie anteriore sufficientemente grande da posizionare tutte le prese di pressione necessarie, nonostante il raggio di dimensioni abbastanza piccole. La condizione di tangenza tra calotta e raccordo crea una superficie sferica, che in sezione si presenta come un arco di circonferenza di 140° (vedi Fig. 1.5), su cui è possibile installare i trasduttori di pressione. La scelta delle dimensioni del cilindro è tale da consentire l'inserimento dei cavi per la trasmissione dei segnali, dalle prese di misura fino all'unità dedicata alla loro elaborazione.

La parte terminale della sonda, invece, è stata disegnata semplicemente in modo tale da minimizzare la resistenza. L'ingombro totale del sensore è di circa 20 cm.

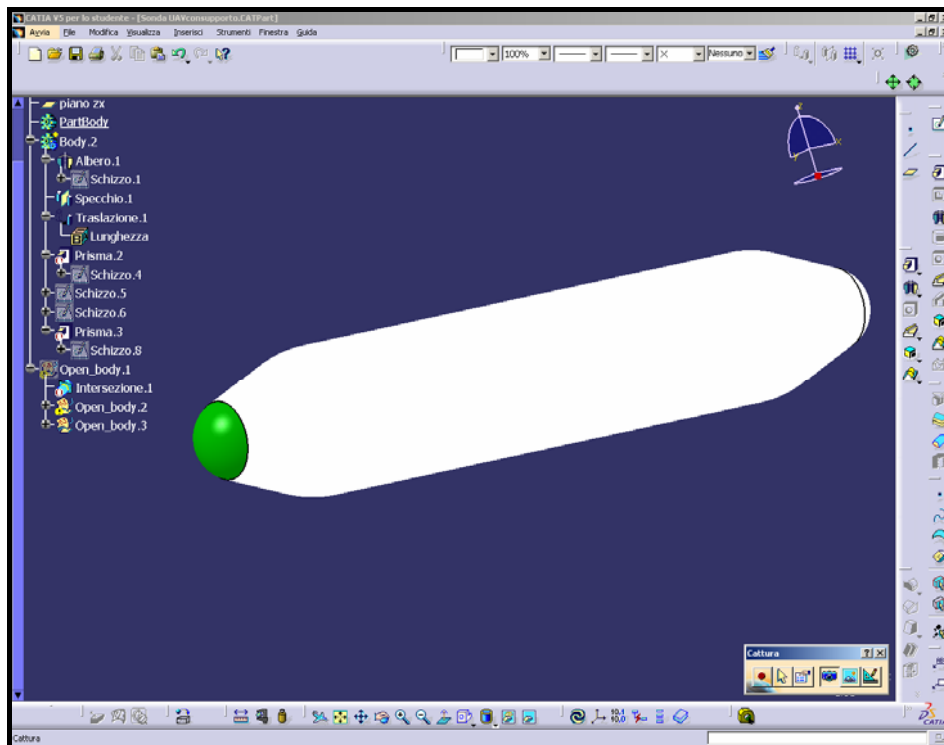


Fig. 1.6 Vista tridimensionale della sonda

Il supporto che permette il collegamento tra la sonda ed il velivolo è costituito da una piastra rettangolare per l'ancoraggio con il velivolo e da un corpo verticale che ha il compito di allontanare la sonda dalla fusoliera, in modo che quest'ultima non crei interferenze aerodinamiche che possano influenzare le misure di pressione.

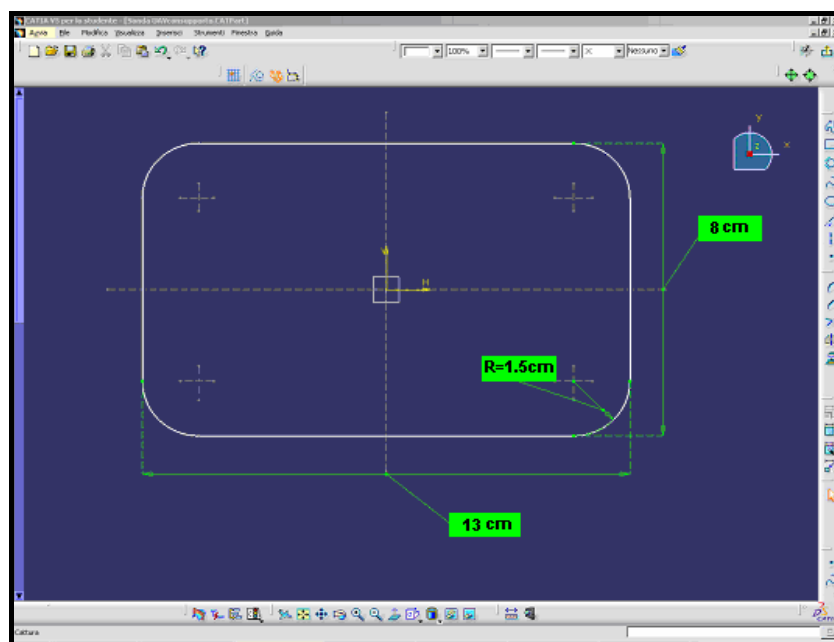


Fig. 1.7 Vista della piastra

La piastra ha il lato maggiore di 13 cm ed il lato minore di 8 cm con dei raccordi agli angoli di raggio 1.5 cm, inoltre presenta uno spessore di 0.4 cm e viene ancorata alla fusoliera tramite quattro spine (vedi Fig. 1.7). Alla piastra è collegato un supporto verticale a forma di prisma con un'altezza di 9.8 cm, con asse verticale inclinato di 45 gradi rispetto alle facce della piastra, sul quale è posizionata la sonda. Il supporto verticale è stato realizzato in modo da minimizzare l'ingombro del cablaggio, ed inoltre in modo da minimizzare resistenza ed interferenza aerodinamica, Fig. 1.8. In sezione (Fig. 1.9) il supporto verticale si presenta come un ellissoide di diagonale maggiore 6 cm e minore 1 cm, sulle due estremità inferiori ha un raggio di curvatura di 0.28 cm mentre lungo i lati maggiori la curvatura aumenta arrivando ad un raggio di 10 cm nel centro.

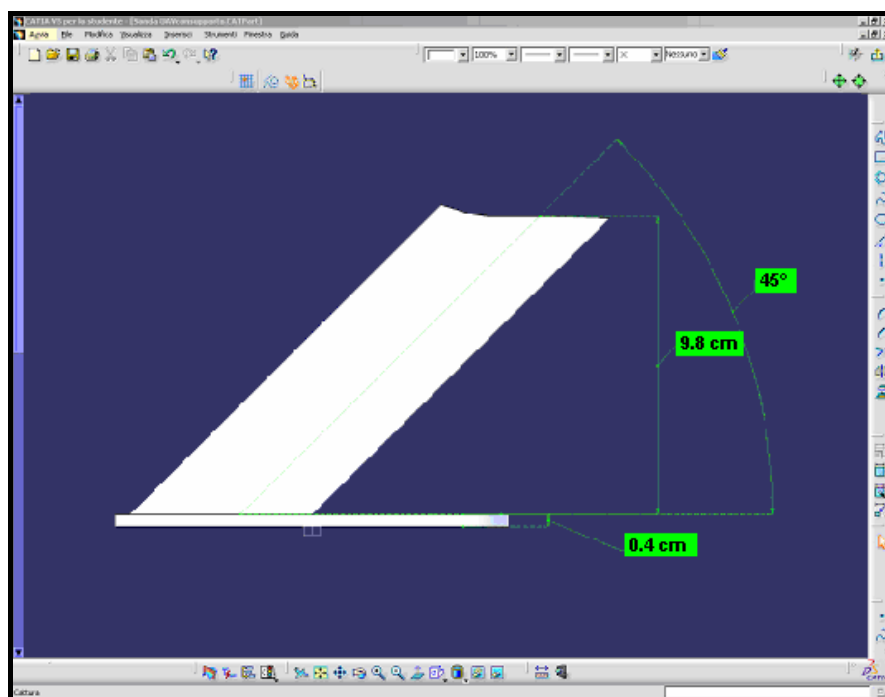


Fig. 1.8 Vista supporto verticale

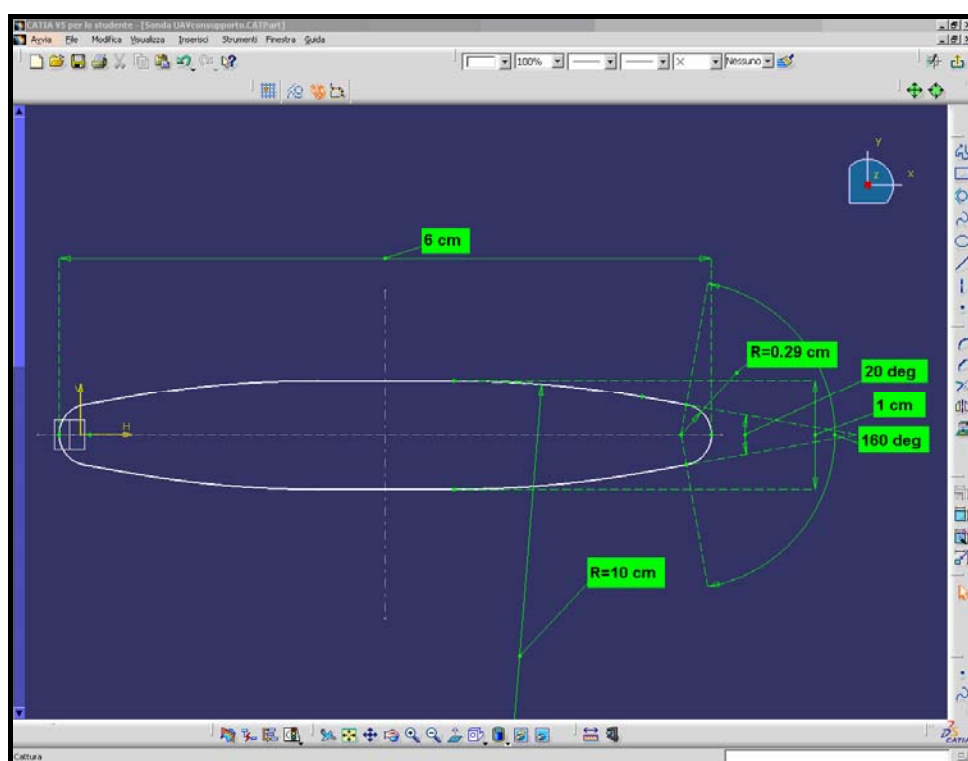


Fig. 1.9 Sezione supporto verticale

Rispetto alla piastra il centro della sezione della base del supporto è posizionato ad una distanza di 3.6 cm dal lato corto e 4 cm dal lato lungo, come mostrato in *Fig. 1.10*. Una volta scelti tutti i parametri della geometria è stata fatta la generazione tridimensionale del modello completo da poter utilizzare successivamente per l'analisi CFD, *Fig. 1.11*.

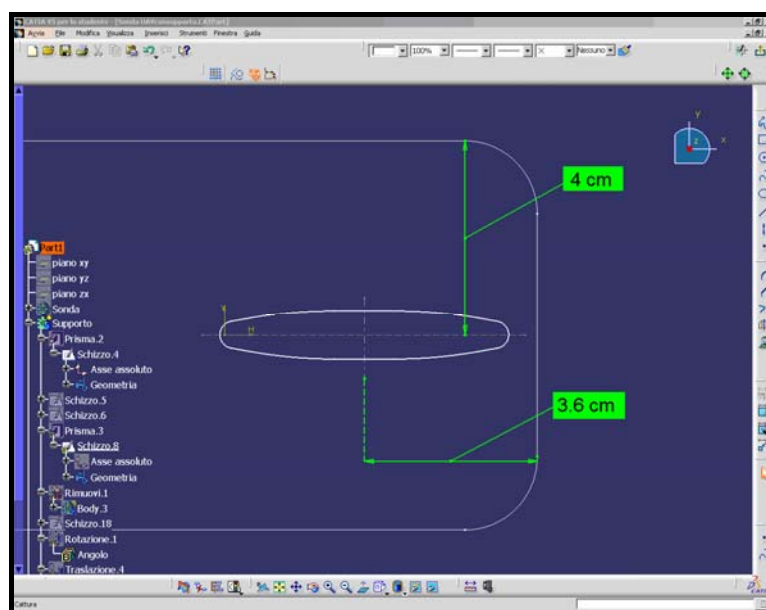


Fig. 1.10 Giunzione tra i supporti

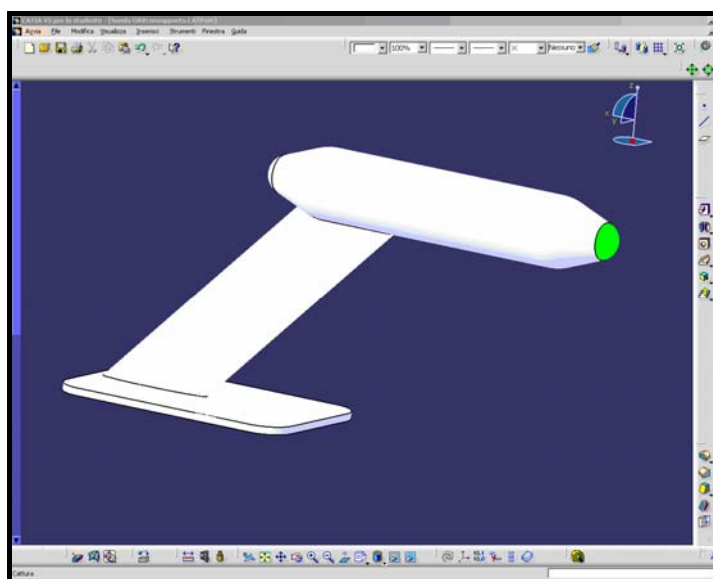


Fig. 1.11 Vista tridimensionale della sonda con supporto

2 Generazione della Mesh

2.1 Introduzione

Una volta definita la geometria della sonda, si è passati alla sua caratterizzazione aerodinamica. Questa fase è stata molto importante per ottimizzare la posizione dei fori d'acquisizione delle pressioni ed ha condotto alla definizione di un database preliminare, preso a riferimento per lo sviluppo degli algoritmi di ricostruzione dei dati aria basati sull'uso di reti neurali.

Tutte le analisi fluidodinamiche sono state eseguite su un modello semplificato della sonda in condizioni “*stand alone*”, sfruttando le simmetrie geometriche e rimandando alla fase di calibrazione in galleria del vento lo studio degli effetti d'interferenza con il velivolo.

Si è quindi trascurato il supporto ed eliminato tutta la parte terminale del corpo principale della sonda stessa, in quanto questa non presenta prese di misura delle pressioni. Queste sono posizionate solamente sulla calotta sferica (in verde *Fig. 2.1*) di dimensioni ridotte rispetto a tutta la sonda. Il modello semplificato utilizzato per l'analisi CFD si presenta come in *Fig. 2.1*.

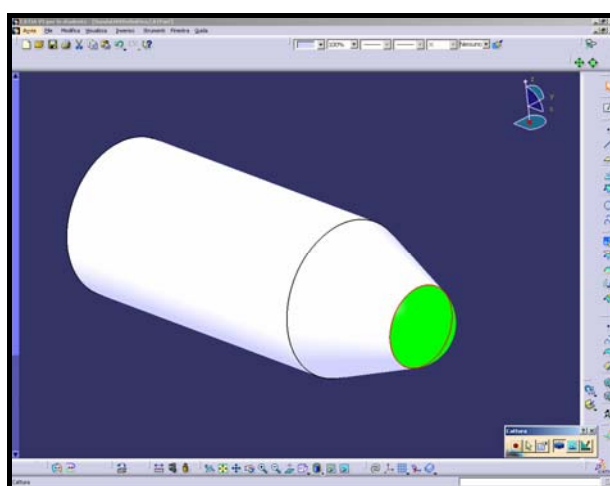


Fig. 2.1 Modello semplificato della sonda

2.2 Descrizione del software *Gambit*[®]

Lo schema di Fig. 2.2 riporta il flusso logico, utilizzato per fare l'analisi fluidodinamica: la geometria solida creata con il *software CATIA*[®], è stata importata nel programma *Gambit*[®] per la generazione della *mesh* o griglia di calcolo. Tale griglia è stata sviluppata per l'analisi fluidodinamica eseguita da *Fluent*[®]. Quest'ultimo è un *software* commerciale basato sul metodo agli elementi finiti per la soluzione di problemi di fluidodinamica [1].

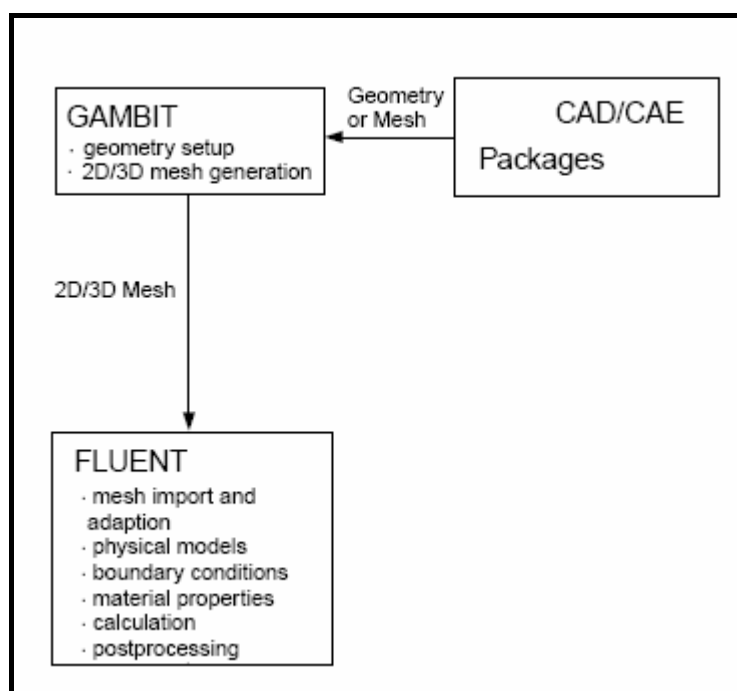


Fig. 2.2 Flusso logico dell'analisi CFD

Il *software Gambit*[®] (versione utilizzata ver.2.2.30) realizza *mesh* di modelli per analisi di fluidodinamica computazionale. *Gambit*[®] riceve l'*input* attraverso la sua *Graphical User Interface* (GUI), che rende la procedura di *meshing* e di assegnazione delle varie condizioni al contorno del modello molto semplice da utilizzare e facilmente adattabile ad una vasta gamma d'applicazioni.

Va osservato che la *mesh* da realizzare deve rappresentare non la sonda ma il flusso d'aria che la investe; quindi partendo dalla sua geometria si sono realizzati dei veri e propri box nei quali è stato ricavato in negativo la sonda stessa. Il modello semplificato è stato dunque utilizzato per generare questi contenitori, dei quali in seguito verranno caratterizzate in dettaglio le superfici con le relative condizioni al contorno, per effettuare l'analisi fluidodinamica il più possibile attinente alla realtà.

La GUI è fondamentalmente suddiviso in tre sezioni: una per la manipolazione della geometria importata, una per la creazione ed il *setup* della *mesh* ed una per la definizione delle condizioni al contorno.

La fase di manipolazione della geometria ha inizio con l'importazione del nostro modello, in particolare si verifica che la geometria non presenti discontinuità ed imperfezioni, come ad esempio linee aperte che genererebbero errori durante la creazione dei volumi. Successivamente si è passati alla definizione della grandezza e del numero di box esterni, che risulta di notevole importanza per questo tipo di analisi.

Infatti realizzando contenitori troppo piccoli si incorre nel rischio di creare delle interferenze tra il flusso che scorre sul modello e quello imperturbato sul contorno del box più esterno; al contrario, contenitori troppo grandi avrebbero bisogno di tempi di calcolo elevati. Quindi si è dovuto trovare un compromesso tra queste due esigenze ed inoltre la scelta delle dimensioni del box esterno è stata fatta in modo tale che l'aria lontano dal profilo risulti imperturbata, ovvero di tipo *free air*. Poichè il nostro modello presenta delle dimensioni massime d'ingombro di 10 cm di lunghezza e 3.6 cm di diametro e che presenta due piani di simmetria xy e xz (Fig. 2.3), come box esterno abbiamo considerato un cilindro che presentasse le seguenti caratteristiche:

- ❖ a monte, 6 volte la lunghezza del cilindro
- ❖ a valle, 15 volte la lunghezza del cilindro
- ❖ diametro pari a 45 volte il diametro del cilindro

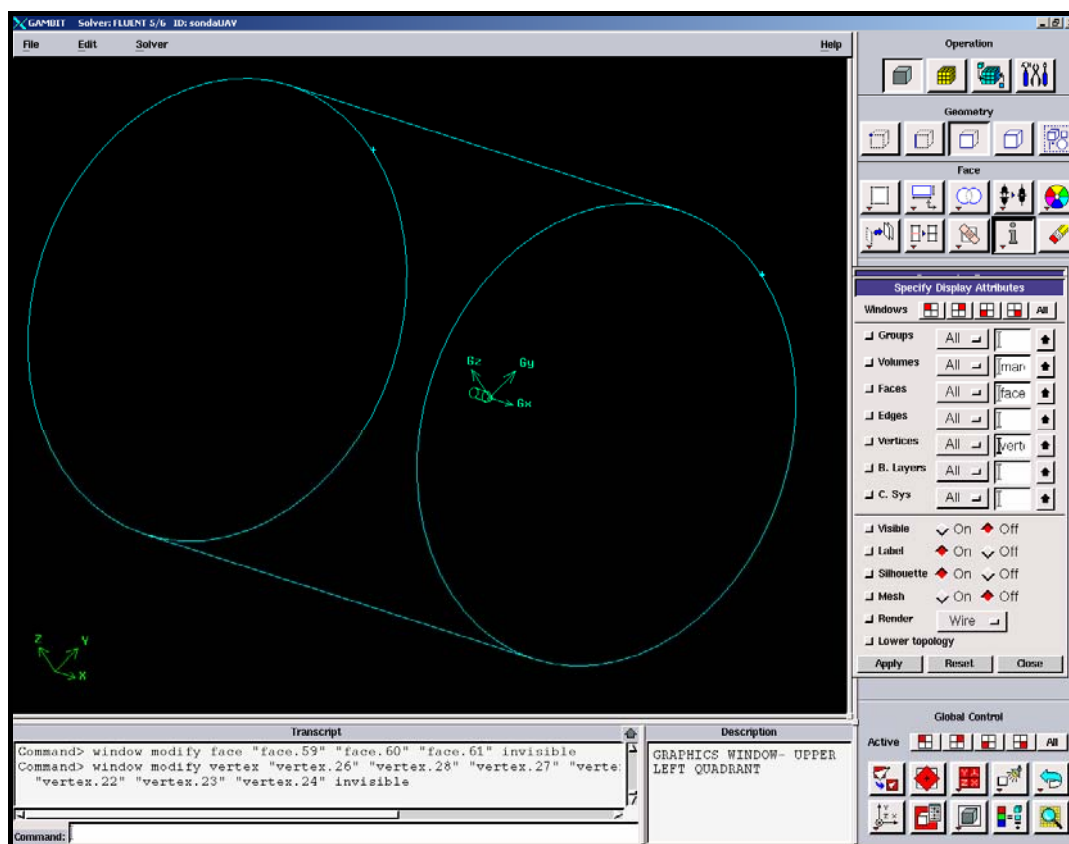


Fig. 2.3 Visualizzazione del box esterno

Dalla Fig. 2.3 si può osservare la grossa differenza di dimensioni tra il modello e il *box* esterno: questo fa sì che il flusso lontano dalla sonda e vicino al *box* stesso sia imperturbato.

Per ottimizzare l'analisi, si è dovuto fare in modo che la *mesh* fosse di dimensioni molto piccole in prossimità della sonda, per avere una notevole precisione di calcolo, e più grossa allontanandoci da essa, poiché l'accuratezza del calcolo non è più necessaria e quindi è possibile diminuire il numero delle celle, riducendo in questo modo anche i tempi di calcolo.

A tal scopo sono stati creati una serie di *box* intermedi: si tratta di 9 scatole concentriche che seguono la forma della sonda nelle sue vicinanze, vedi Fig. 2.4, ed allontanandoci da essa hanno una forma cilindrica con la parte anteriore a calotta sferica, Fig. 2.5. A tali *box* verranno applicate *mesh* di superficie e di

volume con dimensioni differenti. Infatti, partendo da un'opportuna *mesh* superficiale, *Gambit*[®] automaticamente estrapola i volumi con cui riempire i *box*, permettendoci di controllare la crescita di questi elementi dal contorno del corpo fino al *box* esterno. Il numero di celle aumenta vicino al corpo e viceversa diminuisce sul *box* esterno.

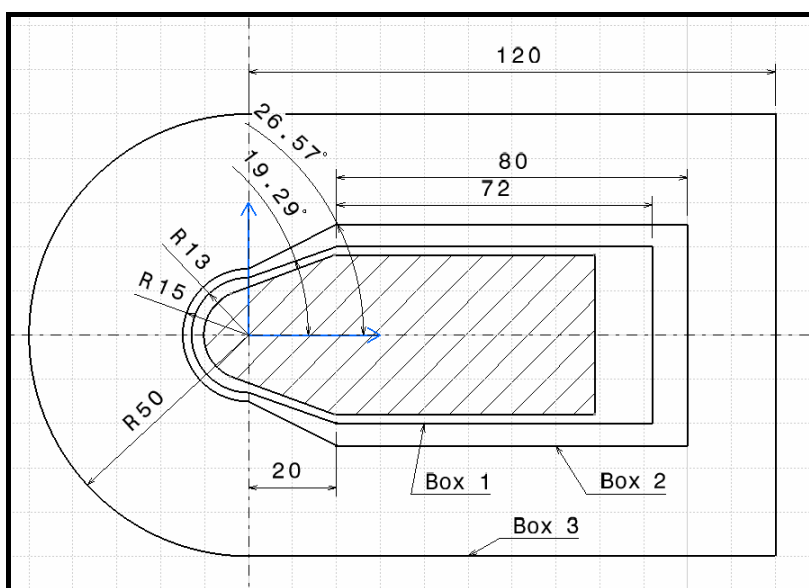


Fig. 2.4 Visualizzazione dei box più vicini al modello

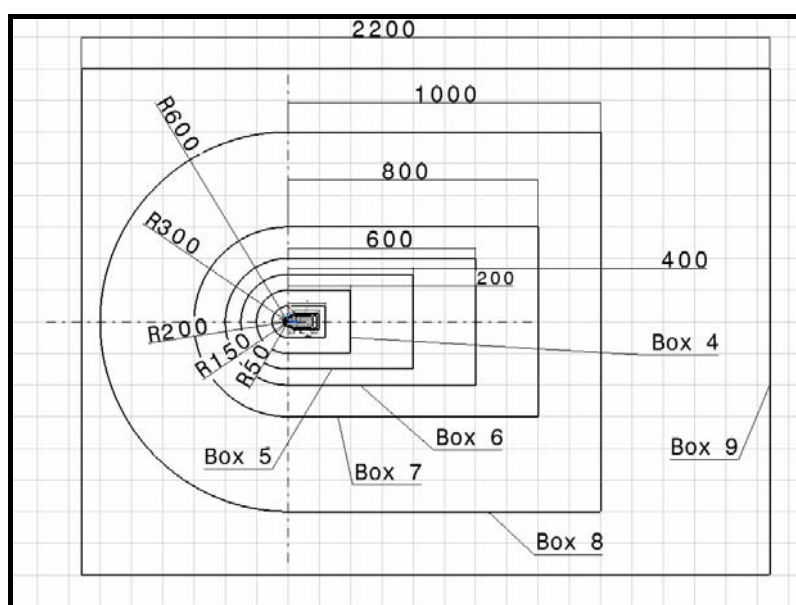


Fig. 2.5 Visualizzazione dei box più esterni

Una volta costruiti i contenitori, *Fig. 2.6*, si è proceduto alla realizzazione della *mesh* utilizzando i comandi dell'interfaccia grafica GUI.

Come già detto in precedenza il primo problema che si è dovuto affrontare è stato quello di creare una griglia abbastanza piccola in modo tale da assicurare una buona precisione nell'analisi fluidodinamica che verrà effettuata in seguito.

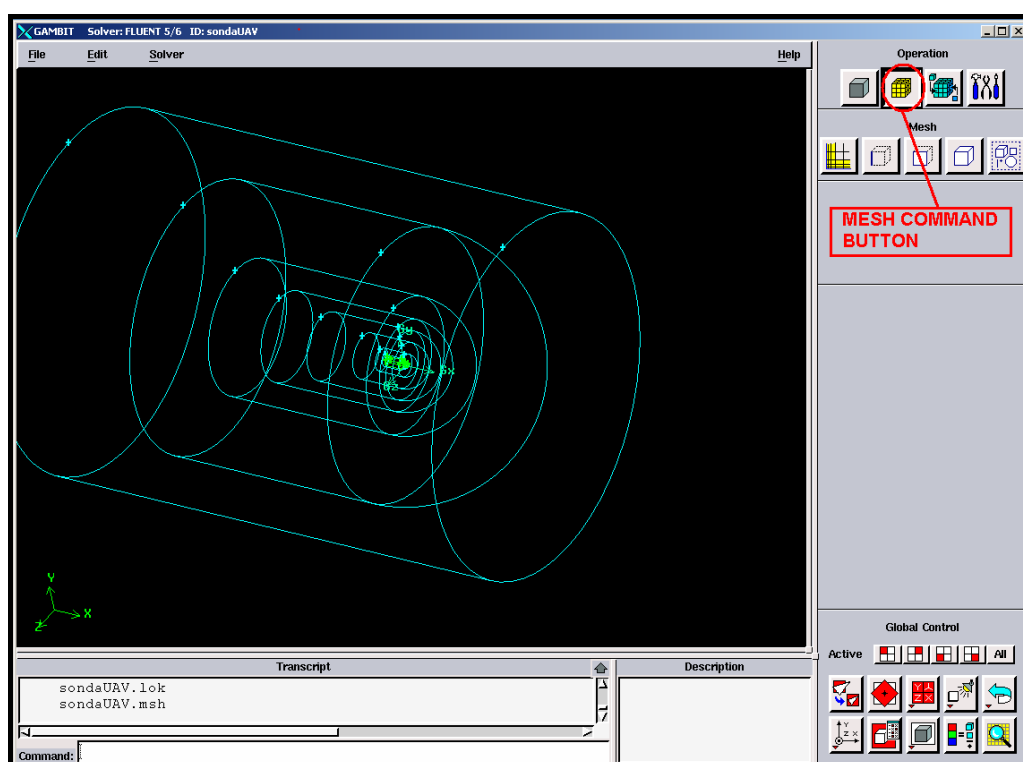


Fig. 2.6 Visualizzazione tridimensionale dei box

Il *software* presenta tre differenti sezioni dedicate rispettivamente alla *mesh* di linea, di superficie e di volume. Ogni sezione presenta una serie di comandi che rendono il *setup* della griglia molto semplice ed intuitivo [11].

Il procedimento consiste nel delineare la griglia o *mesh* superficiale, imponendo una dimensione voluta; e successivamente estrapolare da questa *mesh* quella di volume: tale procedimento viene ripetuto per ogni superficie da definire. La *mesh* di superficie è costituita da numerosi elementini ed il *software* gestisce i vertici di

ognuno di essi come nodi. *Gambit*[®] crea i nodi sulle facce in accordo ai parametri specificati: “schemi della *mesh*”, “spaziatura dei nodi” e “opzioni della *mesh* superficiale”, Fig. 2.7.

Attraverso “schemi della *mesh*” vengono specificati la forma degli elementi ed il tipo di modello per la generazione della griglia. Il *software* consente di creare una griglia con elementi di forma quadrilatera, oppure triangolare o in prevalenza formata da quadrilateri e completata da elementi di forma triangolare.

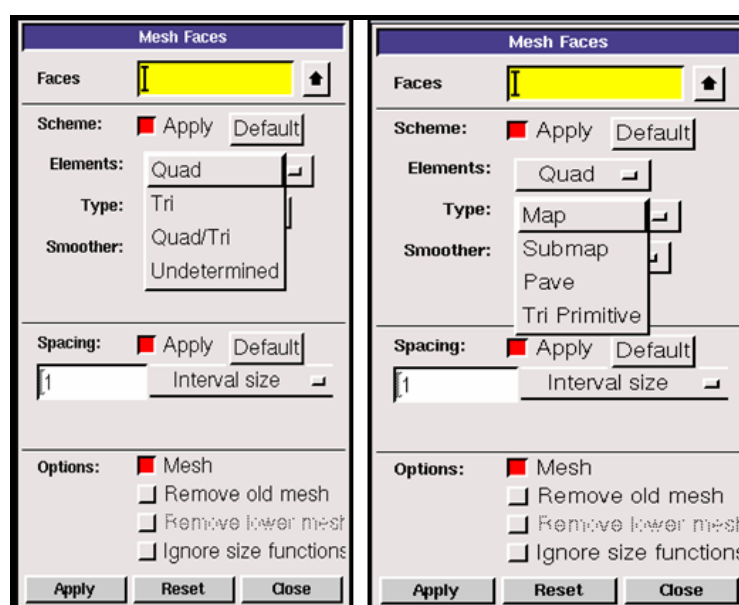


Fig. 2.7 Menu “Mesh Faces”

Ognuna di queste impostazioni di forma è associata ad una o più modelli per la generazione della griglia. Il programma dà la possibilità di scegliere cinque tipi di modelli elencati di seguito:

- MAP crea una griglia regolare e strutturata.
- SUBMAP divide una faccia *unmappable* in tante regioni *mappable* in cui poi genera una griglia regolare e strutturata
- PAVE crea una griglia non strutturata.

- TRI-PRIMITIVE divide le facce con tre lati in tre quadrilateri creando all'interno di ognuno una *mesh mappable*.
- WEDGE-PRIMITIVE crea elementi radiali su facce a forma di cuneo.

Naturalmente non sempre la forma dell'elemento è applicabile al modello specificato, come mostrato in *Fig. 2.8*.

	Elements		
Type	Quad	Tri	Quad/Tri
Map	X		X
Submap	X		
Pave	X	X	X
Tri Primitive	X		
Wedge Primitive			X

Fig. 2.8 Applicabilità degli elementi di superficie al modello di riempimento

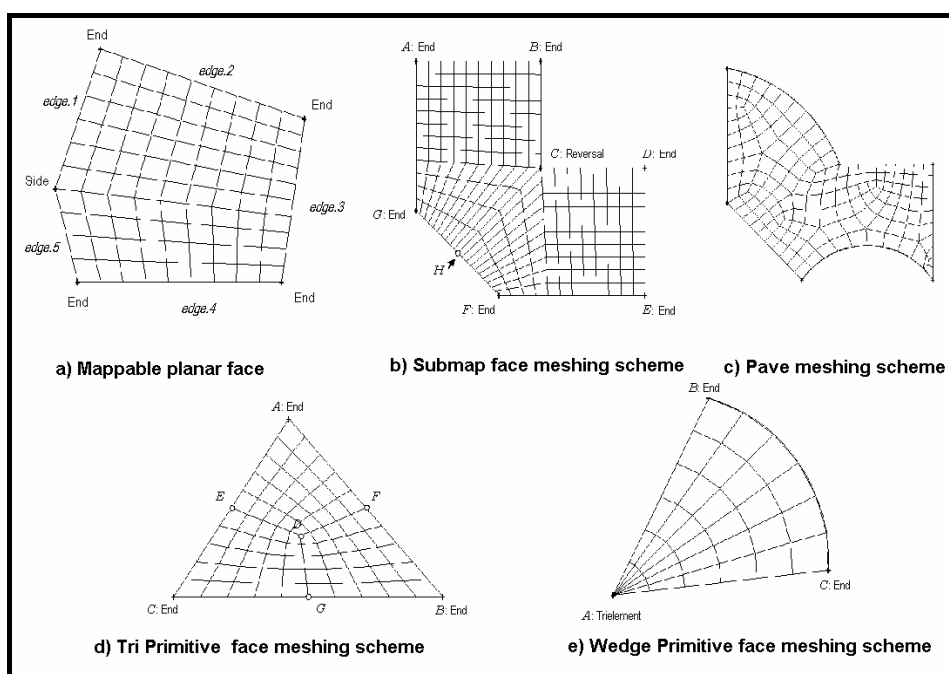


Fig. 2.9 Raffigurazione dei modelli di riempimento della mesh

Una volta selezionata la forma dell'elemento da usare ed il modello per la costruzione della griglia, per completare la *mesh* bisogna impostare la spaziatura tra i nodi.

Quest'ultima è funzione di tre grandezze fondamentali che sono il rapporto di lunghezza dell'intervallo, la lunghezza dello spigolo ed il numero d'intervalli.

I comandi per impostare la spaziatura dei nodi (vedi Fig. 2.10) sono:

- *Interval Count*
- *Interval Size*
- *Shortest Edge (%)*

Selezionando la prima opzione va inserito come *input n* il numero di intervalli in cui va diviso lo spigolo della faccia su cui va costruita la *mesh*, creando un numero di nodi *m* uguale a

$$m = n + 1$$

Il secondo comando invece serve per inserire da tastiera la lunghezza dell'intervallo minimo *d* in maniera tale che il numero di intervalli totali *n* sia

$$n = \frac{L}{d}$$

con *L* la lunghezza dello spigolo.

Usando la terza opzione bisogna fornire al programma, come nel caso precedente, la lunghezza dell'intervallo questa volta espresso in forma di percentuale della lunghezza del lato. Da qui il *software* calcola

$$d = \left(\frac{x}{100} \right) \cdot L_{\min}$$

con *x input* e *L_{min}* la lunghezza dello spigolo più corto della faccia di cui si sta facendo il *meshing*.

Fatto ciò, *Gambit*[®] automaticamente applica a tutti gli spigoli associati alla faccia le caratteristiche selezionate.

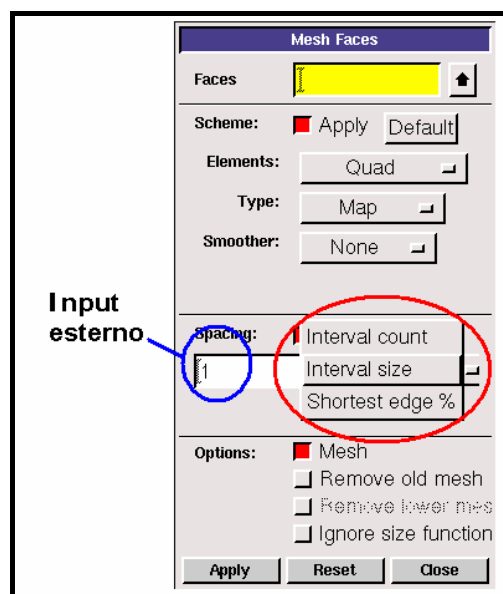


Fig. 2.10 Inserimento della spaziatura tra i nodi

Una volta definiti i parametri della *mesh* di superficie, si è proseguito nella creazione di quella di volume. Come nel caso precedente, *Gambit*[®] presenta una sezione apposita, come mostrato in Fig. 2.12, per definire le caratteristiche principali della *mesh* attraverso i seguenti punti:

- Selezione dei volumi
- Scelta degli schemi della *mesh*
- Definizione della spaziatura dei nodi della *mesh*
- Mesh options

Dapprima si selezionano i volumi nei quali inserire gli elementi tridimensionali della *mesh* di volume, successivamente si specifica lo schema della *mesh* impostando due parametri che sono il tipo di elementi da usare ed il tipo di modello per la generazione della *mesh*.

Il tipo di elementi che possono essere usati per generare il volume sono di diverse forme: tetraedrici, esaedrici, piramidali. Il *software* presenta tre diverse opzioni per settare il tipo di celle:

- Hex
- Hex/Wedge

➤ Tet/Hybrid

La prima opzione riempie il volume usando solo elementi di tipo esaedrico, la seconda include elementi d'angolo che servono come giunzione là dove non si riesce ad inserire un elemento esaedrico. L'ultima opzione utilizza principalmente elementi di tipo tetraedrico aggiungendo, a seconda della grandezza o delle caratteristiche del volume, elementi piramidali, esaedrici e d'angolo (vedi Fig. 2.11).

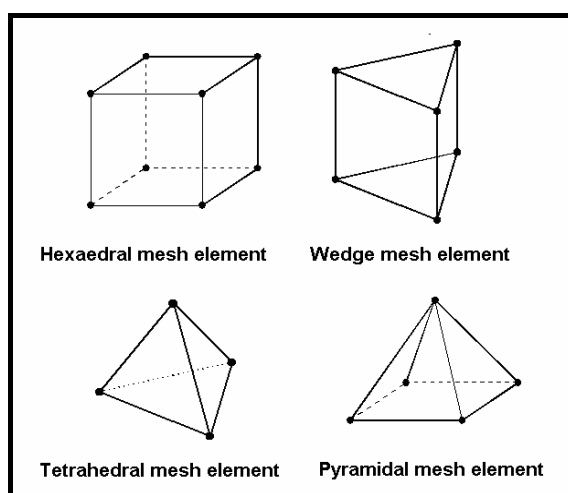


Fig. 2.11 Visualizzazione tridimensionale degli elementi di volume

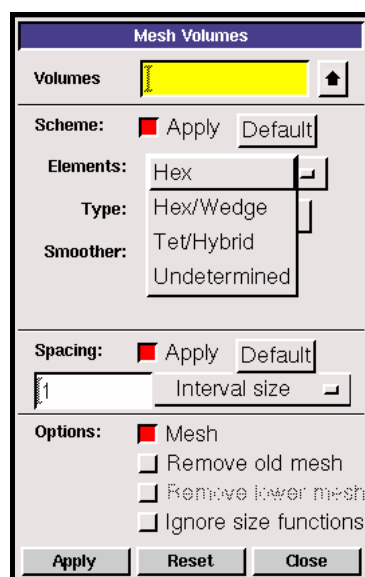


Fig. 2.12 Menu “Mesh Volumes”

Per l'impostazione degli schemi e degli algoritmi di *meshing*, vedi Fig. 2.13, c'è la possibilità di scegliere tra le seguenti opzioni:

- MAP crea una griglia strutturata con elementi esaedrici.
- SUBMAP divide una zona *unmappable* in zone *mappable* in cui crea griglie strutturate con elementi esaedrici.
- TET PRIMITIVE divide il volume in quattro regioni esaedriche in cui crea *mesh* strutturate.
- COOPER estrapola i modelli della *mesh* superficiale in una *mesh* di volume.
- TET/HYBRID crea specificatamente una *mesh* composta principalmente da elementi tetraedrici
- STAIRSTEP crea una *mesh* esaedrica regolare e dove può inserisce volumi con la stessa forma del volume di riferimento.

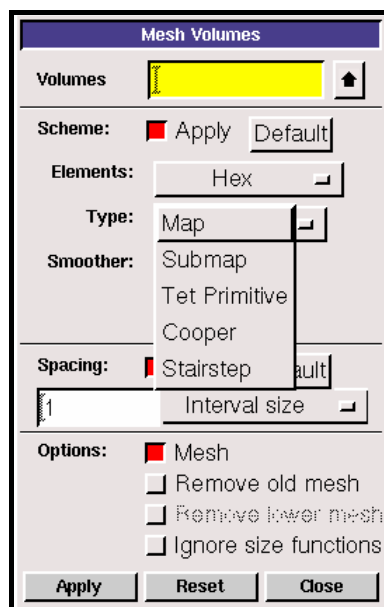


Fig. 2.13 Scelta del modello di riempimento

Come nel caso della *mesh* superficiale anche qui non tutti i tipi di elemento si possono applicare al modello scelto per il riempimento della griglia, come visualizzato in Fig. 2.14.

Una volta scelti la forma degli elementi da usare per creare la *mesh* di volume ed il modello per la generazione della stessa, si è passati a definire l'ampiezza della spaziatura tra i nodi. Il *software* presenta le stesse opzioni d'inserimento che si erano viste in precedenza per l'impostazione della *mesh* di superficie.

	Elements Option		
Type Option	Hex	Hex/Wedge	Tet/Hybrid
Map	X		
Submap	X		
Tet Primitive	X		
Cooper	X	X	
TGrid			X
Stairstep	X		

Fig. 2.14 Applicabilità degli elementi di volume al modello di riempimento

Definiti tutti questi parametri il programma automaticamente costruisce nella maniera più idonea la *mesh* di volume, adattandola alle impostazioni settate sulle superfici comuni. Inoltre all'interno di *Gambit*[®] sono presenti una serie di opzioni che è possibile utilizzare per migliorare la raffinatezza della *mesh*.

2.3 Realizzazione della mesh

Nel presente paragrafo verranno evidenziati i settaggi effettuati durante il nostro lavoro per realizzare la *mesh*. Nella realizzazione della *mesh* superficiale sono stati utilizzati tutti elementi di forma triangolare: altri elementi infatti ci avrebbero creato difficoltà di posizionamento sulle parti sferiche. Il modello di griglia utilizzato è di tipo *Pave*.

Per la spaziatura tra i nodi della griglia si è fornito dall'esterno direttamente la misura dell'intervallo utilizzando l'opzione *Interval Size*.

Inoltre, per rendere il modello più accurato e ridurre al minimo il numero delle celle utilizzate, si è dovuto ingrandire la dimensione degli elementi superficiali, dalla calotta sferica della sonda alla superficie cilindrica che costituisce il corpo posteriore della sonda stessa, poiché questa parte non ci interessa ai fini del nostro lavoro. Per la parte anteriore della sonda è stata utilizzata una spaziatura tra i nodi di 0.03 cm come visualizzato in *Fig. 2.15*.

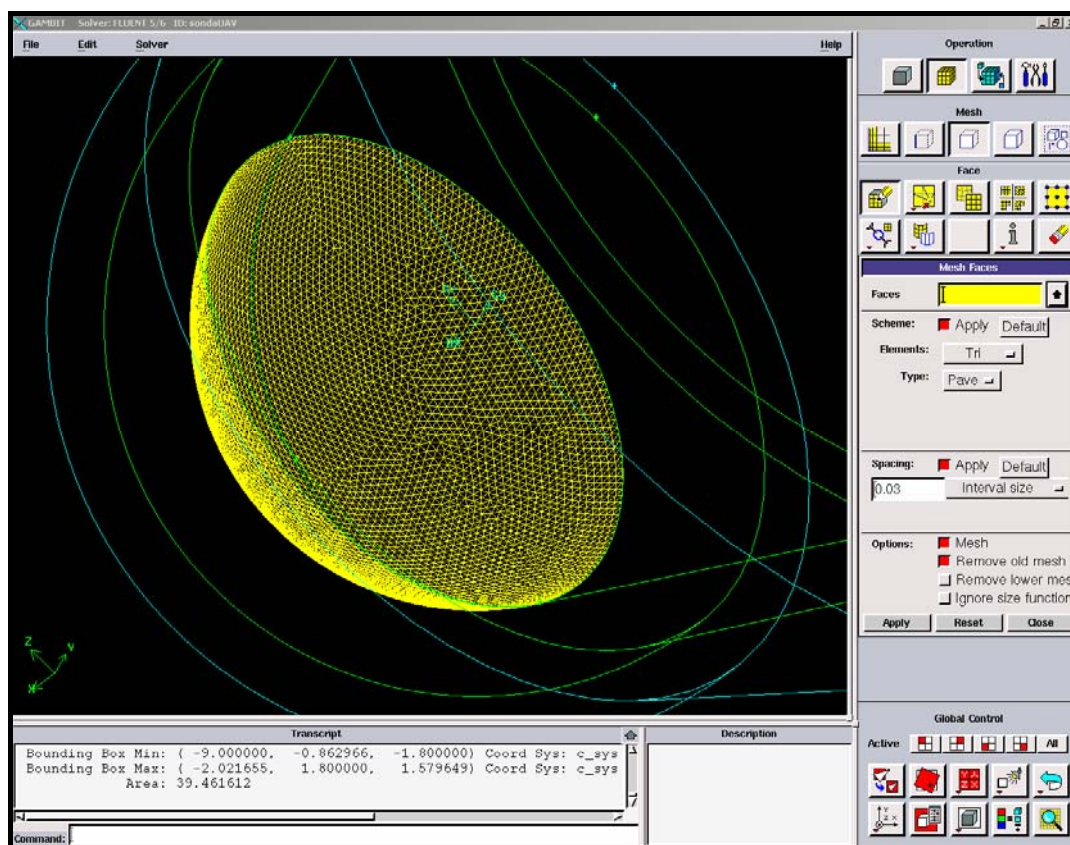


Fig. 2.15 Mesh di superficie della parte anteriore della sonda

Spostandosi indietro sul corpo della sonda si è incrementato il valore della spaziatura fino a 0.044 cm sulla parte tronco-conica e 0.07 cm sul cilindro posteriore.

Dalla Fig. 2.17 si osserva che, nelle linee di contorno in comune tra una superficie e l'altra, il *software* mantiene i nodi creati durante il primo *meshing* (spaziatura di 0.03cm) e a partire da questi genera la seconda *mesh* superficiale (con spaziatura di 0.044 cm). Ciò vale per tutte le linee di contorno comuni a due generiche superfici.

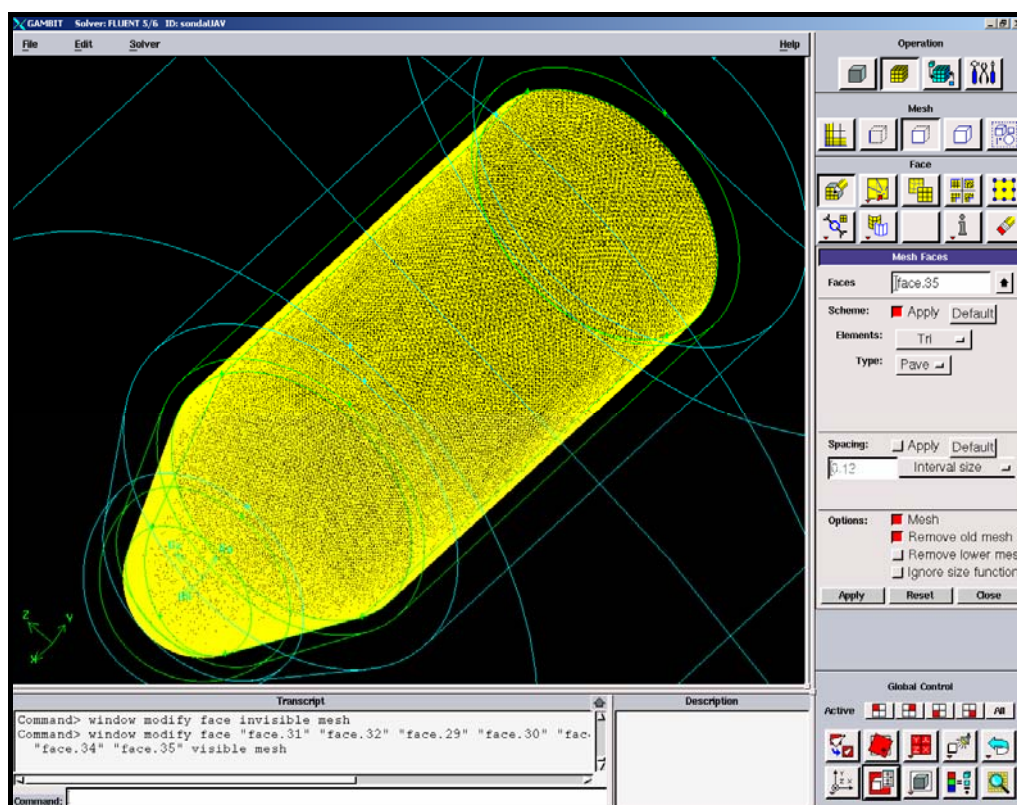


Fig. 2.16 Mesh superficiale del modello

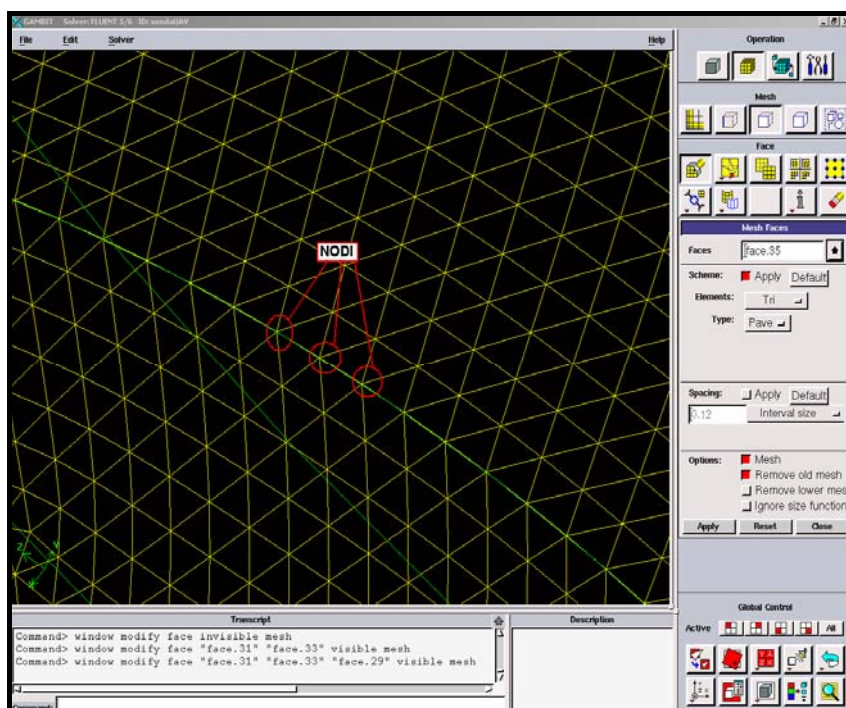


Fig. 2.17 Visualizzazione dei nodi lungo una linea di contorno tra due superfici

Una volta realizzata la *mesh* sulle pareti della sonda, Fig. 2.16 siamo andati a caratterizzare tutte le superfici dei *box*. Come modello di griglia si è utilizzato lo stesso che applicato alla sonda, per non avere problemi durante la creazione della *mesh* di volume, cambiando solo la spaziatura tra i nodi. Anche durante questa fase, allontanandoci dalla sonda, si è cercato di aumentare sempre di più la grandezza degli elementi della griglia, aumentando la spaziatura tra un nodo e l'altro.

Per i primi due *box* intorno alla sonda, la scelta della spaziatura è stata particolarmente delicata, in quanto si richiedeva una precisione elevata. Come detto in precedenza la forma di questi due *box* è simile a quella della sonda.

Sulla calotta del primo si è impostata la spaziatura in modo da creare una griglia con nodi distanti 0.1 cm l'uno dall'altro: sul corpo cilindrico, invece, si è aumentata la spaziatura arrivando a 0.12 cm come visualizzato in Fig. 2.18 e Fig. 2.19.

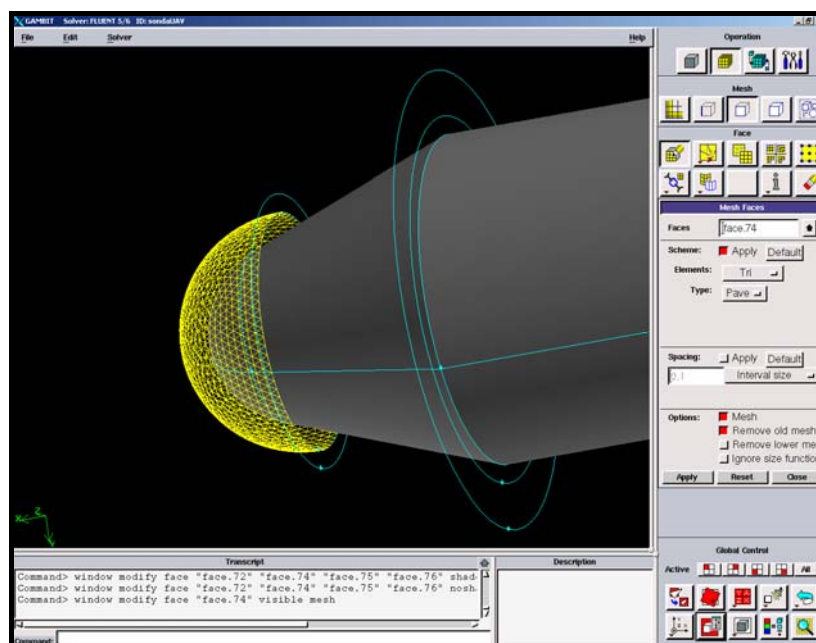


Fig. 2.18 Visualizzazione della mesh superficiale sul primo box

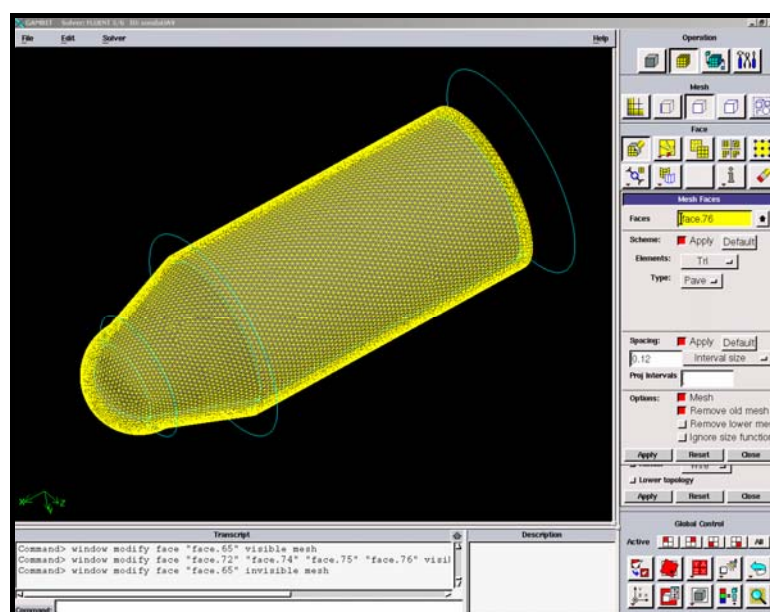


Fig. 2.19 Visualizzazione della mesh superficiale completa del primo box

Lo stesso si è fatto sul secondo *box* arrivando ad una distanza tra i nodi di 0.2 cm sulla calotta sferica e di 0.22 cm sul corpo cilindrico, come visualizzato in Fig. 2.20 e Fig. 2.21.

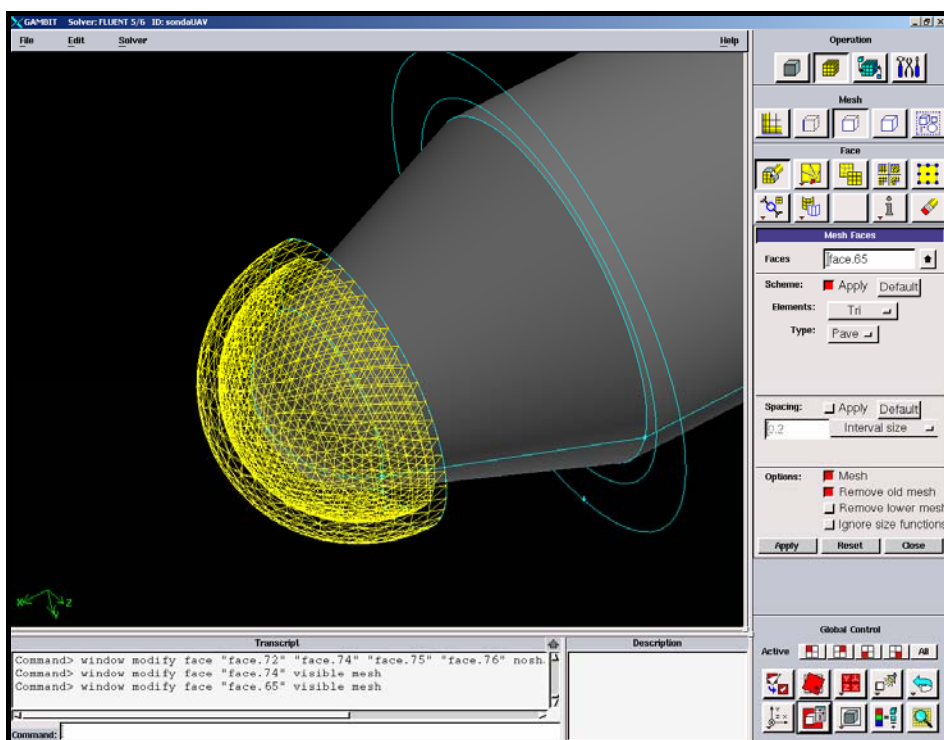


Fig. 2.20 Visualizzazione della mesh superficiale sul secondo box

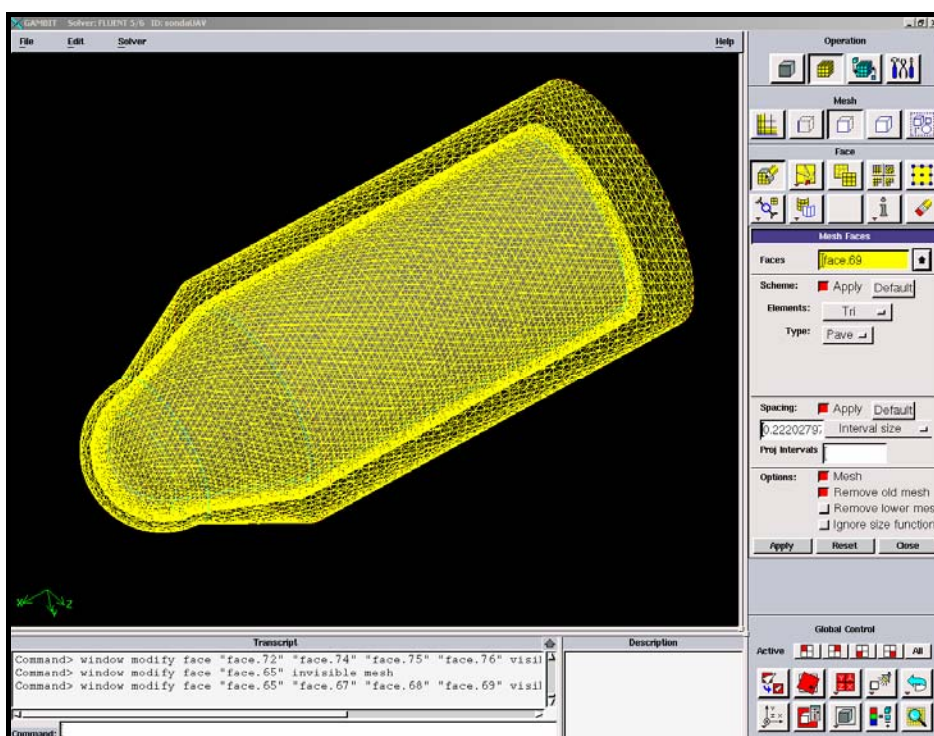


Fig. 2.21 Visualizzazione della mesh superficiale completa del secondo box

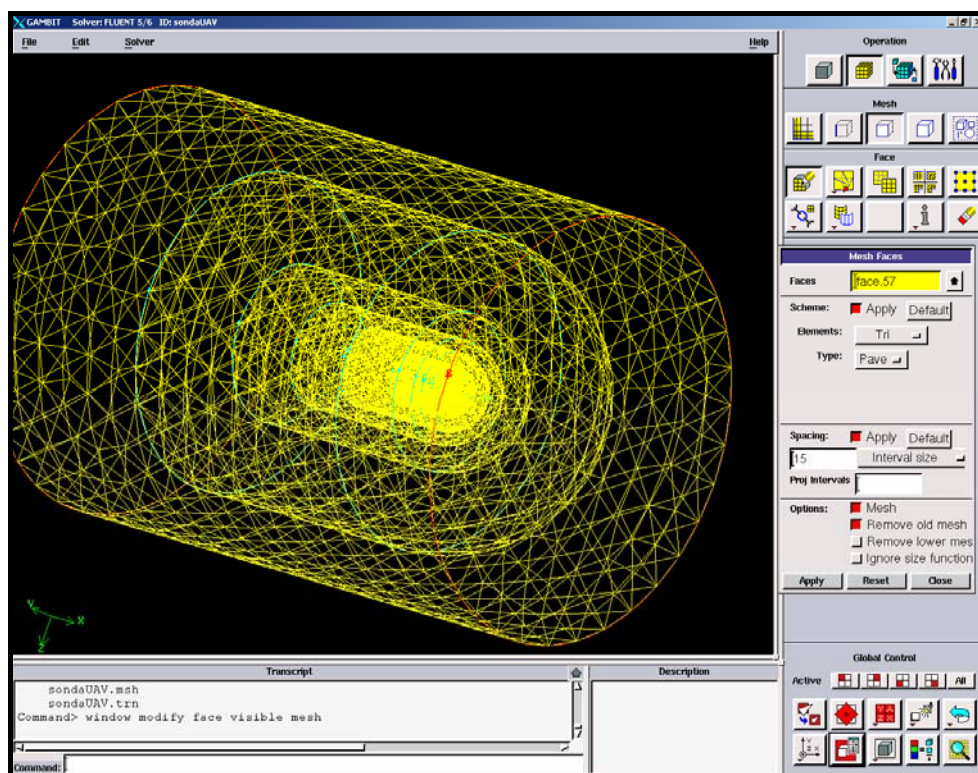


Fig. 2.22 Visualizzazione della mesh superficiale del modello completo

Muovendosi verso l'esterno si è incrementato ancora lo spazio tra le celle arrivando a 15 cm sul box più esterno. In Fig. 2.22 è visualizzato il modello completo con le *mesh* di superficie.

Una volta caratterizzate tutte le superfici sono stati riempiti i *box* con elementi di volume spaziando i nodi tramite l'*interval size*.

Per quanto riguarda il tipo di elementi si sono utilizzati, per tutti i volumi presenti nel modello, elementi di forma tetraedrica. Come schema di riempimento si è utilizzato invece il *Tgrid*, che, come già detto, dove non riesce ad inserire tali elementi utilizza anche forme piramidali esaedriche e d'angolo.

I *box* molto vicini alla sonda presentano elementi di volume di piccole dimensioni ed i *box* più esterni elementi di dimensioni più grandi, per cercare di diminuirne il numero e di conseguenza ridurre i tempi di calcolo. Nei primi due *box* adiacenti la sonda, le dimensioni impostate per lo spazio tra i nodi sono state

rispettivamente 0.12 cm e 0.25 cm. Come osservato in precedenza per le linee di contorno della *mesh* di superficie, anche per le superfici di contorno comuni a due volumi il *software* si comporta nello stesso modo (Fig. 2.23).

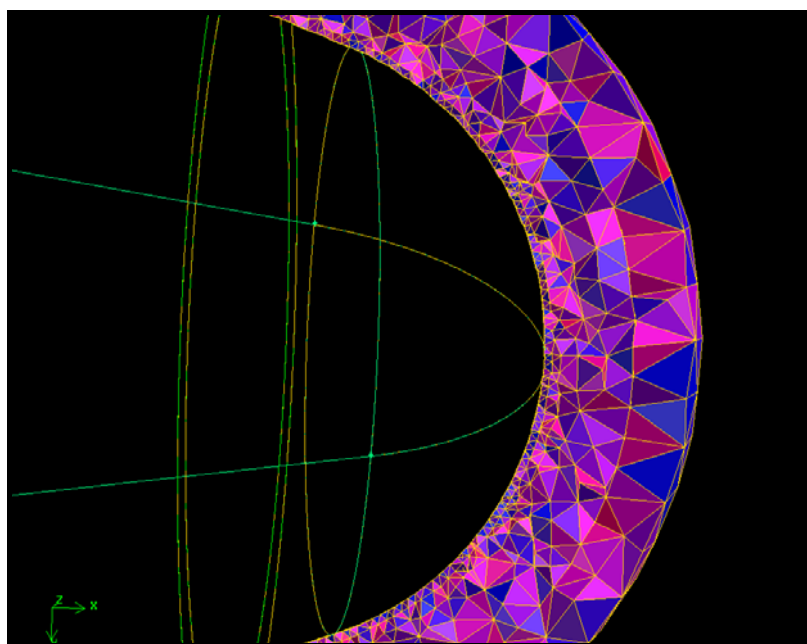


Fig. 2.23 Elementi lungo le superfici di contorno tra volumi

Il *software* crea le celle della *mesh* di volume successiva, sulla superficie di contorno, partendo dalle celle della *mesh* di superficie precedente: questo meccanismo ci permette di controllare zona per zona la crescita degli elementi di volume. Procedendo in maniera analoga si sono riempiti tutti i volumi del modello, raggiungendo, nel *box* più esterno, una dimensione di 15 cm per la spaziatura tra i nodi. Nella Fig. 2.24 è riportata la sezione della *mesh* completa del modello.

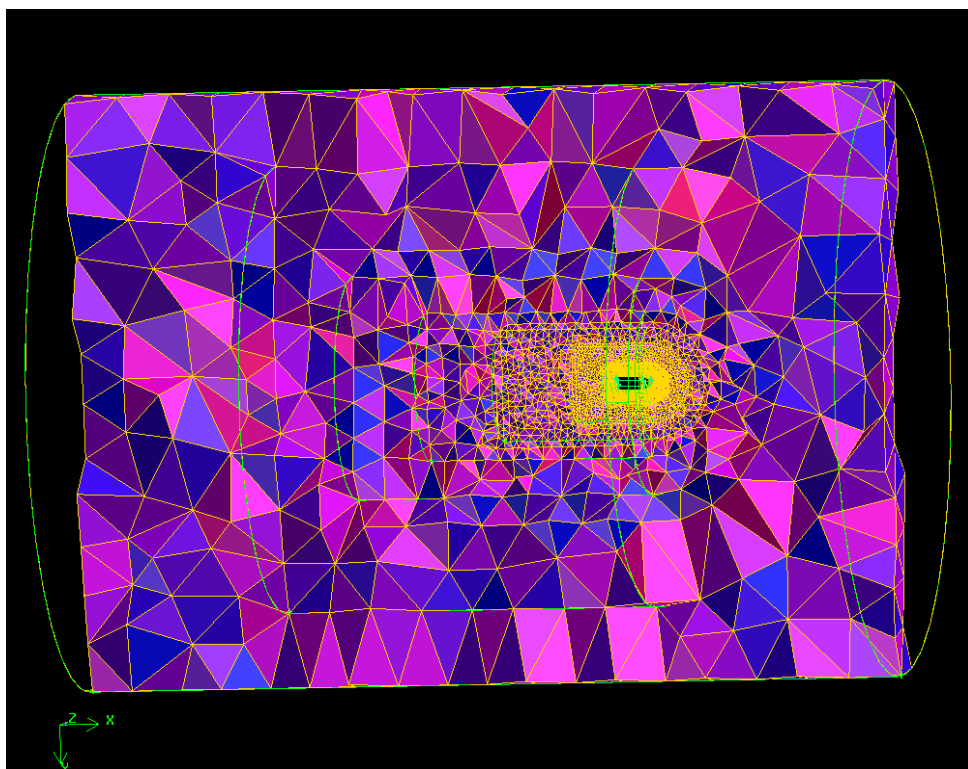


Fig. 2.24 Sezione della mesh di volume del modello

2.4 Condizioni al contorno

Una volta creata la *mesh* si è dovuto impostare condizioni al contorno di ogni zona del modello, operazione necessaria per preparare il modello ad essere esportato in *Fluent*[®].

A questo scopo *Gambit*[®] presenta un apposito comando chiamato *Zone Command Button* che imposta le condizioni al contorno sia delle superfici che dei volumi creati, Fig. 2.25. In particolare in questa fase i volumi presenti nel modello sono stati settati come *Fluid*, e per ogni superficie si è potuto scegliere tra diverse tipologie di impostazione:

- *Wall*
- *Inlet Vent*

- *Pressure Inlet*
- *Interior*
- *Outlet*

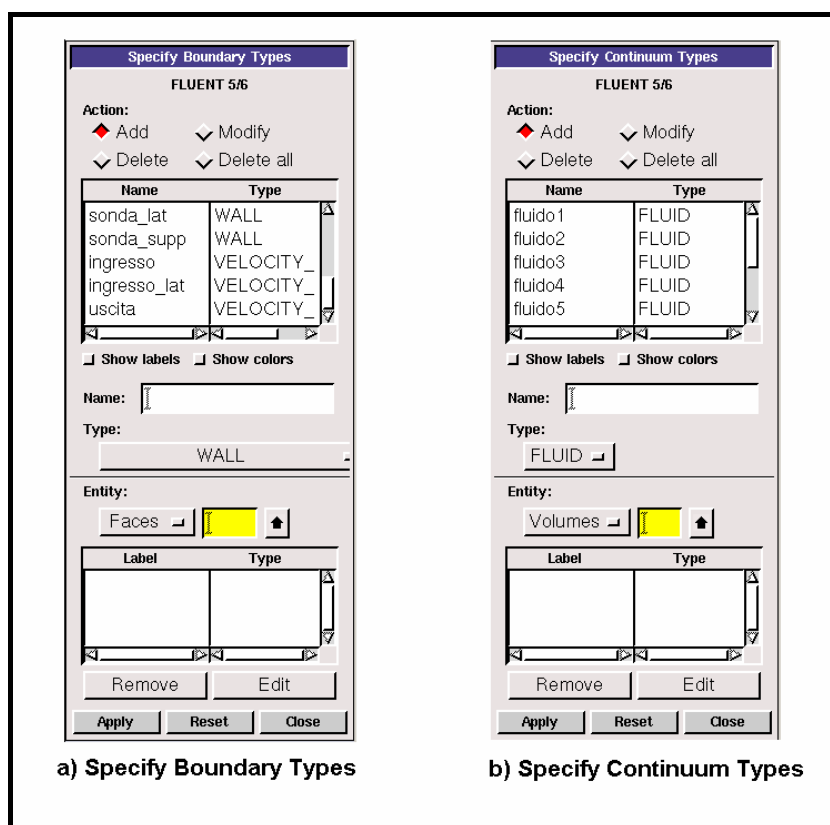


Fig. 2.25 Visualizzazione menu d'inserimento delle condizioni al contorno

Ognuno di questi settaggi fissa sulla superficie uno o più parametri e li mantiene costanti durante la simulazione. Tali impostazioni risultano necessarie per la successiva importazione in *Fluent*[®] e verranno descritte dettagliatamente durante l'analisi CFD.

3 Generazione del database aerodinamico mediante analisi CFD

3.1 Introduzione a Fluent[®]

Una volta terminata la creazione della griglia, il modello completo è stato esportato in *Fluent[®]*, *software* commerciale che permette di simulare attraverso un'analisi di fluidodinamica computazionale, (*Computational Fluid Dynamic*, CFD) il comportamento dinamico dei fluidi in complicati problemi fisici. Tale simulazione consiste nell'elaborazione numerica di sofisticati modelli matematici che descrivono l'evoluzione temporale dei parametri fondamentali del fluido, quali pressione, velocità, temperatura e densità [1].

In questi ultimi anni, grazie soprattutto all'enorme sviluppo delle capacità dei calcolatori, la fluidodinamica numerica ha avuto un'evoluzione tale da renderla, in molti settori dell'ingegneria, uno strumento di supporto indispensabile per la progettazione.

A volte, nonostante l'uso di equazioni semplificate e computer molto veloci, si riesce ad arrivare soltanto a soluzioni approssimative. La validità dei risultati forniti dalla CFD dipende senz'altro da innumerevoli fattori, come verrà evidenziato in seguito, ma una volta ottimizzata la metodologia di calcolo, un modello CFD è in grado di fornire simulazioni molto soddisfacenti, permettendo di eliminare le configurazioni di minor efficienza molto prima della sperimentazione e del progetto di dettaglio. Un altro grande vantaggio offerto dalla CFD è l'indipendenza rispetto al fattore di scala che permette di semplificare notevolmente tutta la fase di impostazione dei parametri fluidodinamici. Ciò permette di ottimizzare l'intero processo di progettazione riducendo enormemente tempi e costi.

La simulazione CFD ha avuto come obbiettivo quello di ottimizzare il posizionamento delle prese di misura ed inoltre ha permesso la definizione di un database preliminare su cui addestrare le reti per la ricostruzione dei dati aria.

La descrizione del moto di un fluido avviene attraverso le equazioni fondamentali della fluidodinamica, che sono rappresentate dall'equazione di continuità della massa (scritta in coordinate cartesiane)

$$\frac{D\rho}{Dt} = 0 \quad (Eq. 3.1)$$

Dall'equazione di conservazione della quantità di moto (spesso indicata come Equazione di Navier-Stokes),

$$\rho \cdot \frac{D\vec{V}}{Dt} = \rho \cdot \vec{G} - \text{grad}p + \mu \cdot \nabla^2 \vec{V} \quad (Eq. 3.2)$$

e l'equazione di conservazione dell'energia

$$\rho \cdot C_p \cdot \frac{DT}{Dt} = \tau_{ij} \cdot \frac{\partial V_j}{\partial x_j} + \frac{DP}{Dt} + \frac{\partial q_i}{\partial x_i} \quad (Eq. 3.3)$$

Dove ρ rappresenta la densità del fluido, μ la viscosità dinamica, C_p il calore specifico a pressione costante, T e P rispettivamente la temperatura e la pressione del fluido, q il calore scambiato per unità di massa mentre G le forze per unità di volume; infine V rappresenta la velocità del fluido e τ_{ij} le tensioni tangenziali di deformazione. Le (3.1-3) rappresentano un sistema di equazioni differenziali alle derivate parziali non lineare e, a causa della sua notevole complessità, non possono in generale essere risolte in forma chiusa.

Spesso la risoluzione delle equazioni di *Navier-Stokes* può essere ottenuta facendo riferimento a ipotesi semplificative sul tipo di fluido e sul tipo di efflusso. Ad esempio, se il fluido è considerato non viscoso si ottengono le equazioni di Eulero, se invece viene ipotizzato un flusso potenziale si ottengono le equazioni di Laplace.

La fluidodinamica computazionale si occupa invece della risoluzione numerica delle equazioni di *Navier-Stokes* scritte nella loro forma completa. Le tecniche risolutive delle (3.1-3) utilizzate dalla CFD sono essenzialmente tre:

- *Direct Numerical Simulation* (DNS)
- *Large Eddy Simulation* (LES)
- *Reynolds Averaged Navier Stokes equations* (RANS)

Nel nostro lavoro è stato utilizzato l'approccio RANS che risulta attualmente il più impiegato per la risoluzione dei problemi di interesse ingegneristico. Con questa metodologia le equazioni di Navier-Stokes vengono risolte considerando le variabili fisiche caratteristiche mediate nel tempo. E' interessante osservare come il costo computazionale diminuisca passando dalla metodologia DNS a quella RANS [12].

Nelle equazioni RANS, tutte le variabili fisiche dipendenti dal tempo sono espresse come somma di due contributi: il valore medio rispetto ad un intervallo di tempo caratteristico e la perturbazione rispetto a tale valore medio.

La tecnica RANS è una metodologia approssimata di risoluzione delle equazioni di *Navier-Stokes*. In particolare, con tale metodo non si valuta l'effetto locale della turbolenza ma piuttosto l'effetto della turbolenza sulle grandezze medie (variabili che, dal punto di vista ingegneristico, sono di maggiore interesse). Va infine sottolineato che la risoluzione delle RANS fornirà risultati tanto più soddisfacenti quanto meglio il modello di turbolenza riuscirà ad approssimare la fisica del problema oggetto di studio.

Si osservi che il dominio del fluido viene discretizzato dal calcolatore col metodo dei volumi finiti in celle elementari di piccole dimensioni (elementi), che vanno a formare la griglia di calcolo (detta *mesh*): ad essa si applicano dei metodi di risoluzione iterativa al fine di risolvere le equazioni del moto.

Tali procedimenti iterativi vengono interrotti una volta raggiunto il grado di accuratezza desiderato. Naturalmente l'importanza del settaggio dei parametri dell'analisi è determinante per una sua corretta realizzazione: un errore potrebbe infatti compromettere la veridicità della simulazione.

3.2 Impostazione delle simulazioni

I passi principali da seguire sono stati rappresentati nello schema in *Fig. 3.1*.

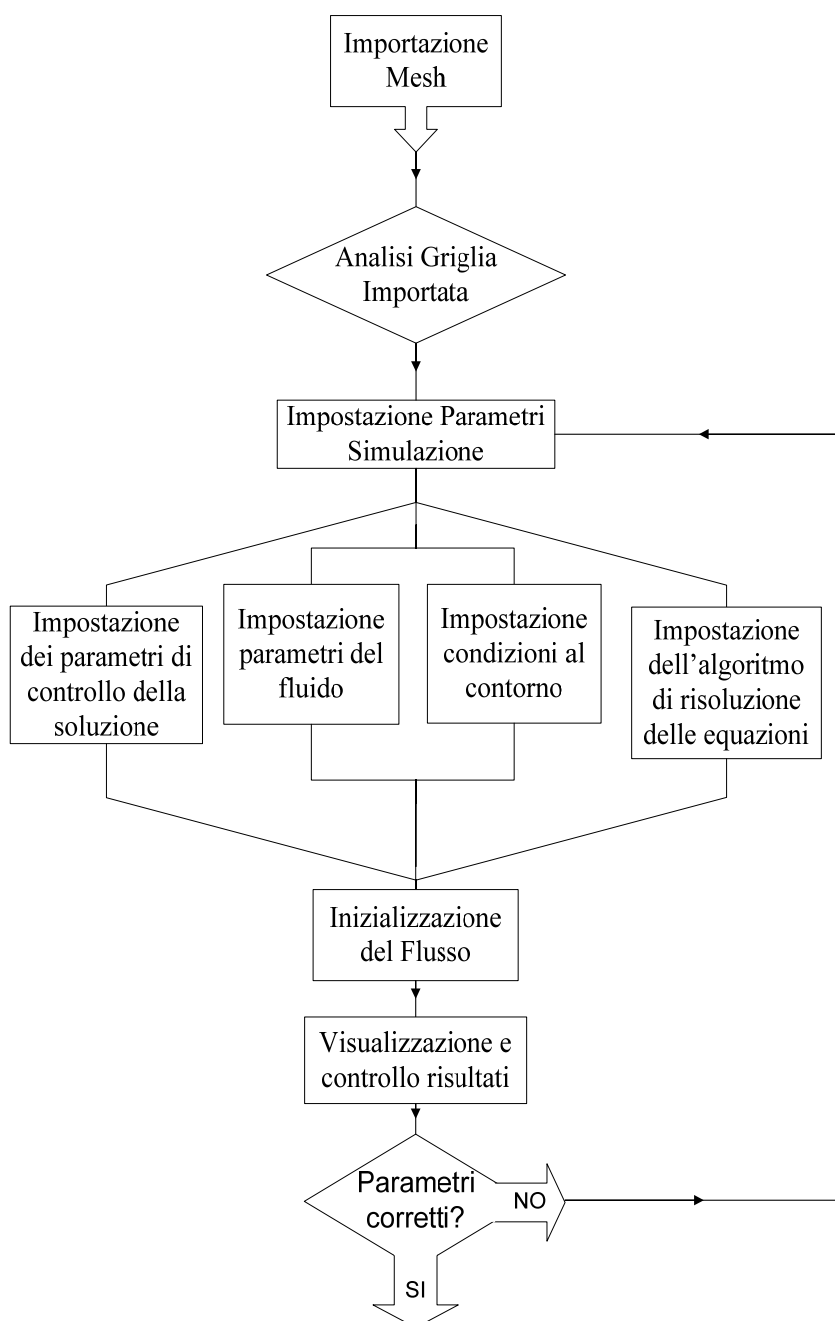


Fig. 3.1 Flusso logico dell' impostazione dei parametri della simulazione

Il primo passo è stato compiuto importando in *Fluent*[®] la griglia, o dominio fisico, creata con *Gambit*[®], controllando che non ci siano errori di geometria e che la scala del modello importato sia corretta.

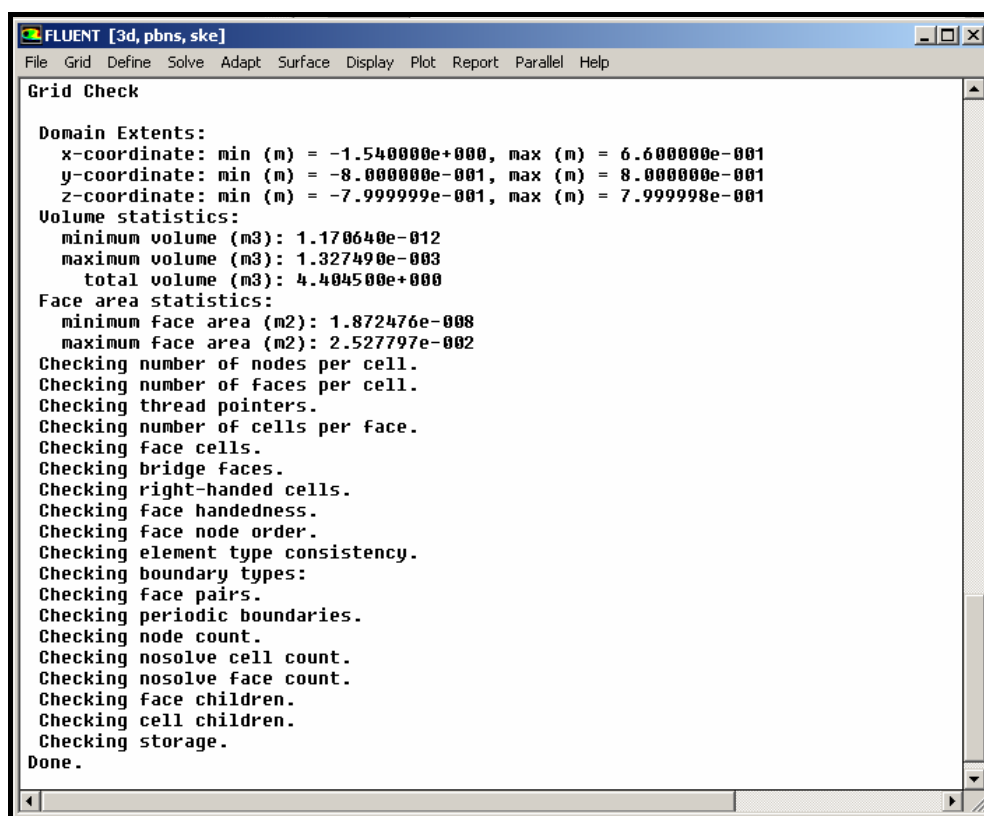


Fig. 3.2 Visualizzazione “Grid Check”

Verificato tutto ciò come si vede in Fig. 3.2, si è passati al settaggio dei parametri del flusso agendo per prima cosa sull'impostazione del tipo di soluzione che vogliamo ottenere.

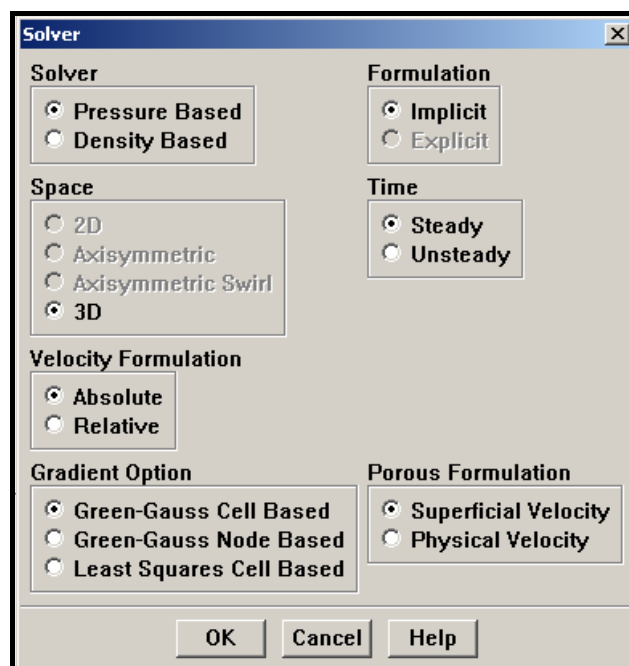


Fig. 3.3 Visualizzazione Menu "Solver"

A riguardo, è stato possibile impostare la dipendenza della soluzione delle equazioni del moto dal tempo ed il punto in cui le varie grandezze vengono calcolate. Dal menu *Solver* quindi si seleziona *Steady* per effettuare un'analisi di tipo stazionario, eliminando così la dipendenza dal tempo ed infine *Green-Gauss Cell Based* dal sottomenu *Gradient Option* per prendere le misure nelle vicinanze del centro della cella come media aritmetica della grandezza in esame.

Successivamente si è selezionato dal menu *Viscous* il tipo di equazioni da usare per caratterizzare i flussi turbolenti. Infatti tali flussi presentano fluttuazioni del campo di velocità che miscelano le quantità trasportate (momento, energia, densità, ecc.) e fanno sì che tali quantità possano variare. Poiché tali fluttuazioni sono piccole e ad alta frequenza è troppo difficile simularle direttamente. Così vengono creati dei modelli semplificati aggiuntivi che prevedono nuove variabili capaci di simulare la presenza della turbolenza. Il modello da noi scelto a tale scopo è il *k - ε Model* per il fatto che si adatta notevolmente alle equazioni *Reynold Averaged Navier Stokes* (RANS) in caso stazionario. Questo modello

inserisce due equazioni nelle due incognite (k , ϵ) dove k rappresenta l'energia cinetica di turbolenza e ϵ il rateo di dissipazione dell'energia cinetica stessa.

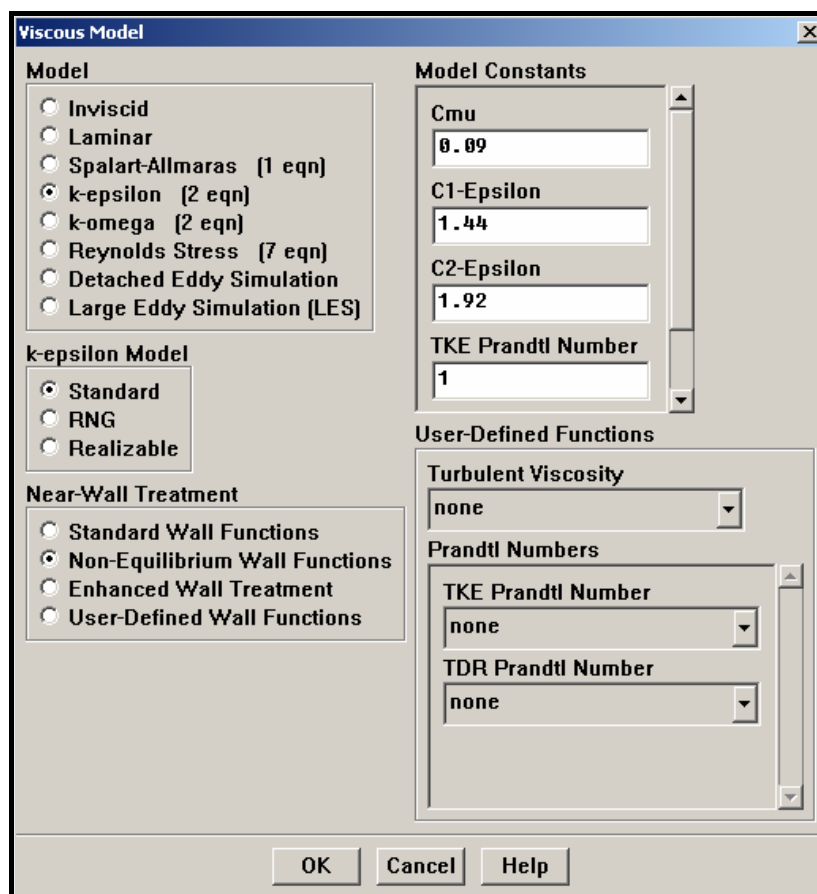


Fig. 3.4 Visualizzazione Menu “Viscous Model”

Fluent[®] ci permette, tramite il menu visualizzato in Fig. 3.4, di selezionare tale modello ed introdurre automaticamente le costanti di cui ha bisogno per risolvere le equazioni aggiuntive a differenza di altri *software* di fluidodinamica computazionale. E' possibile inoltre selezionare come trattare la turbolenza in prossimità delle pareti (nel nostro caso *Non-Equilibrium Wall Function*): su tali superfici infatti la variazione, dovuta alla turbolenza, di tutti i parametri viene impostata a zero perché la *mesh* è talmente fine da poter risolvere il modello attraverso l'uso delle equazioni per flusso laminare.

Successivamente si è scelto il tipo di materiale e le condizioni al contorno di tutte le superfici presenti nel modello.

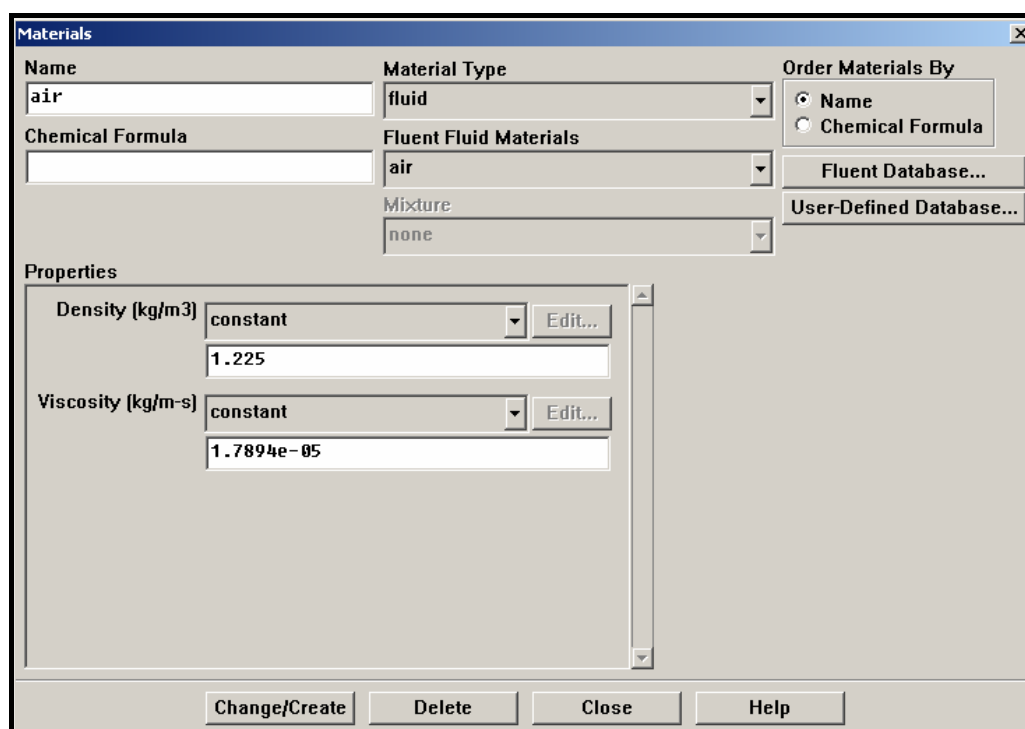


Fig. 3.5 Visualizzazione Menu “Materials”

Come visualizzato in Fig. 3.5 il fluido utilizzato per l’analisi CFD è l’aria con i relativi valori di densità e viscosità.

Per le condizioni al contorno, sulle superfici interne al modello è stata scelta l’impostazione *Interior*; impostazione che fa sì che su tali superfici tutti i parametri del flusso rimangano invariati; le superfici di contorno esterne sono state tutte considerate come *Velocity Inlet*; le superfici del modello come *Wall*, cioè come una parete solida. Per quanto riguarda quest’ultima opzione è bastato imporre che la velocità della parete fosse nulla (ovvero non scorrimento alla parete), lasciando i valori di *default* per la rugosità. Sulle superfici considerate *Velocity Inlet* invece, poiché si è fissato il valore della velocità, oltre a dare modulo e direzione della

velocità del flusso, si è dovuto impostare la turbolenza scegliendo come dati d'ingresso intensità 0.1% e rapporto di viscosità 10 (Fig. 3.6).

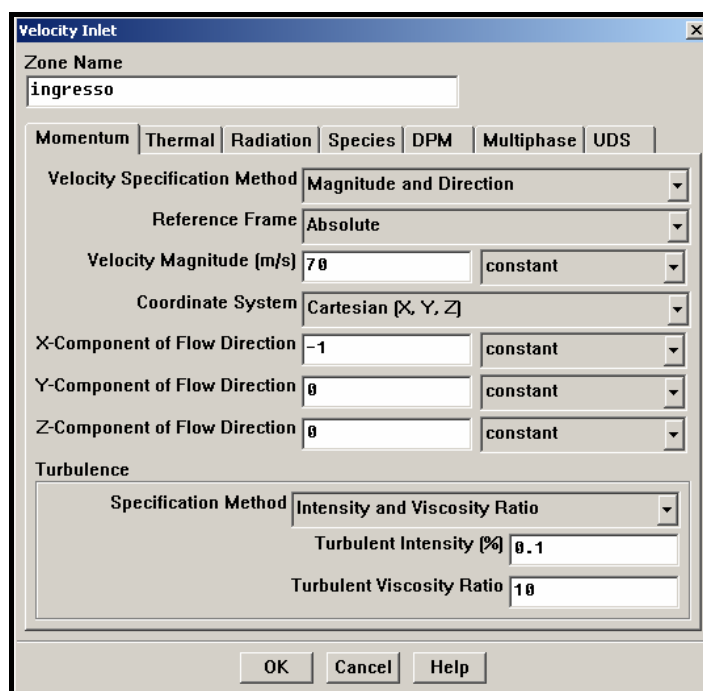


Fig. 3.6 Visualizzazione Menu “Velocity Inlet”

Successivamente è stato impostato il controllo della soluzione (vedi Fig. 3.8), prendendo in considerazione sia le equazioni del flusso che quelle della turbolenza. Inoltre si osservi che per convertire le equazioni di stato in equazioni algebriche risolvibili matematicamente, *Fluent*[®] utilizza una tecnica basata sul controllo della soluzione attraverso degli opportuni volumi detti di controllo. L'integrazione di ogni grandezza viene eseguita per ogni volume riferendosi al centro della cella in considerazione (Fig. 3.7).

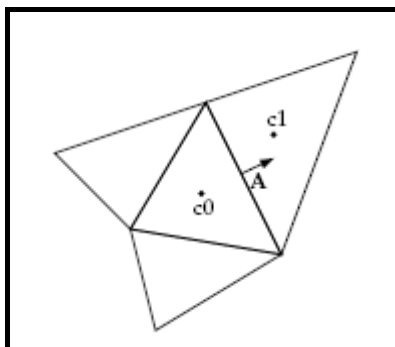


Fig. 3.7 Visualizzazione del generico volume di controllo

Per ricavare i valori di tutte le grandezze lungo i bordi della cella viene fatta un'interpolazione a partire dal centro, usando degli schemi di “upwind”, ovvero la grandezza viene ricavata attraverso un'interpolazione lungo la direzione del flusso [1]. Nel nostro caso per alcune grandezze quali momento, energia cinetica e rapporto di dissipazione dell'energia si è utilizzato il comando “second order upwind” che imposta un'accuratezza del secondo ordine.

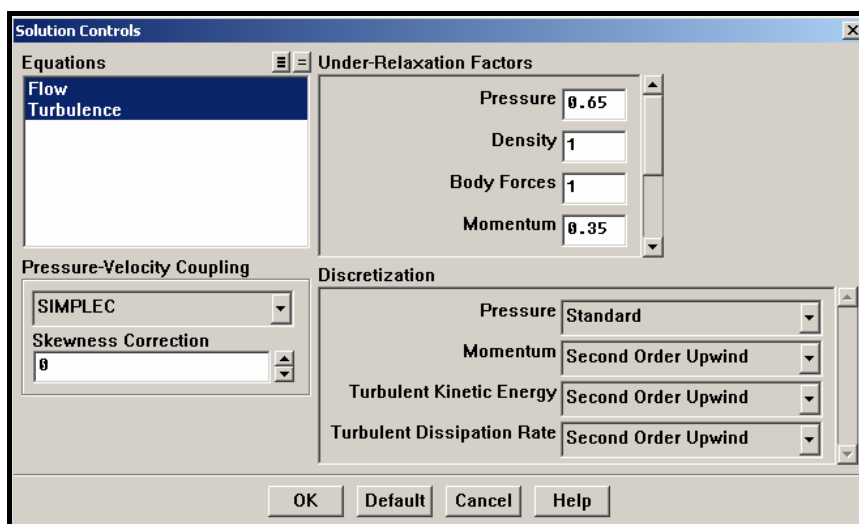


Fig. 3.8 Visualizzazione Menu “Solution Controls”

3.3 Posizionamento dei punti di controllo

Una volta preparata la simulazione, si è passati alla definizione dei punti di controllo, ovvero delle prese di misura delle grandezze di interesse sulla superficie del modello: come mostrato in *Fig. 3.9* si è considerato i quattro archi di circonferenza e su questi sono stati creati 85 punti. Sugli archi principali (archi rosso e blu di *Fig. 3.10*) i punti sono stati posizionati con intervalli di 5 gradi, sui restanti archi invece, essendo questi di interesse minore per il nostro studio, sono stati presi punti spaziati di 10 gradi.

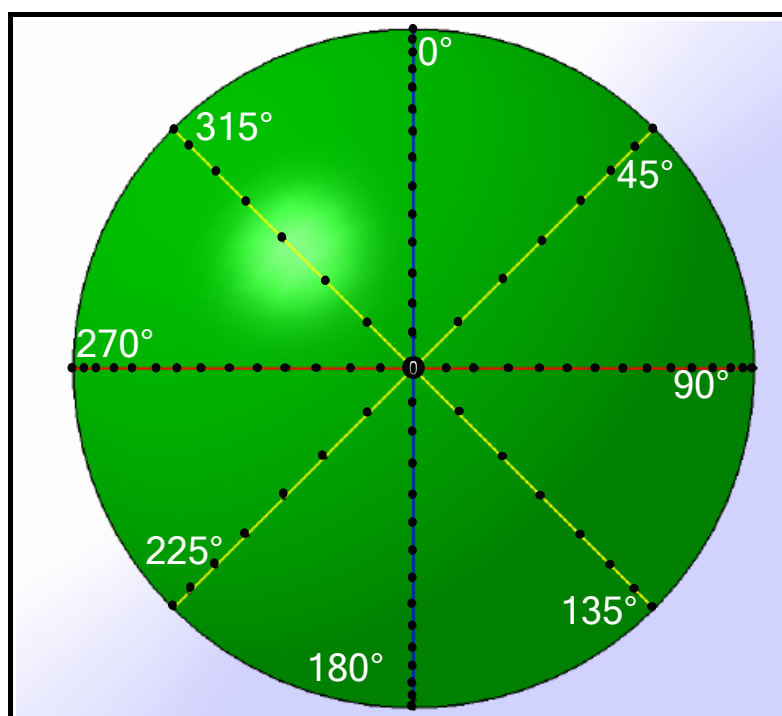


Fig. 3.9 Visualizzazione degli archi sulla sonda

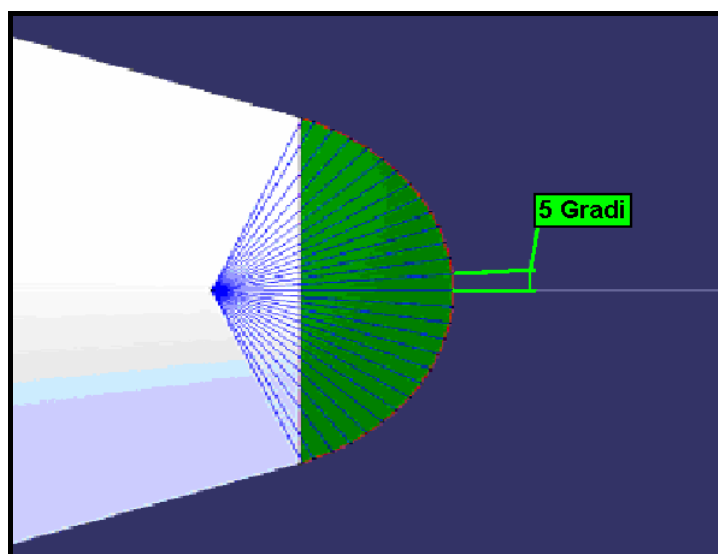


Fig. 3.10 Visualizzazione della spaziatura sugli archi principali

Una volta creati i punti di controllo, come visualizzato in *Fig. 3.11*, sono state avviate le simulazioni costruendo in questo modo in tutto l'involuppo di volo un database preliminare che ci servirà nella seconda parte del nostro lavoro. Le simulazioni sono state effettuate per angoli d'incidenza α che vanno da -5 a $+15$ gradi, angoli di derapata β da -15 a $+15$ gradi e per velocità da 20 m/s a 70 m/s.

Poiché i tempi di simulazione sono piuttosto lunghi si sono presi in esame i principali punti dell'involuppo di volo considerato, intervallati di 5 gradi nell'angolo d'incidenza ed imbardata e di 10 m/s nella velocità; inoltre si è cercato, visionando i dati volta per volta, di sfruttare la simmetria geometrica del nostro modello.

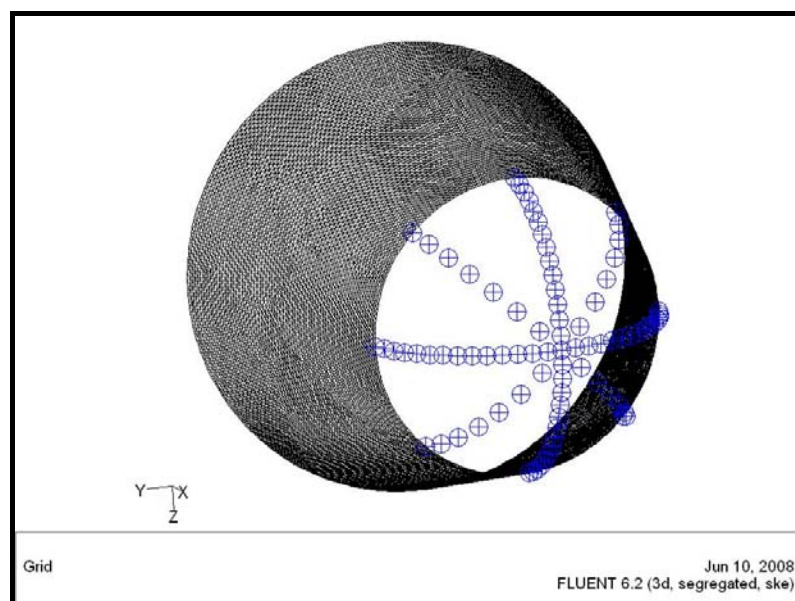


Fig. 3.11 Visualizzazione punti di controllo

3.4 Analisi dei risultati delle simulazioni CFD

Come visualizzato in *Fig. 3.12*, *Fluent*[®] risolve, partendo da dei valori iniziali, contemporaneamente sia le equazioni di continuità che quelle di momento ed energia. Successivamente viene presa in esame l'equazione per il calcolo della turbolenza ed i risultati vengono confrontati con dei valori di riferimento per valutarne la convergenza.

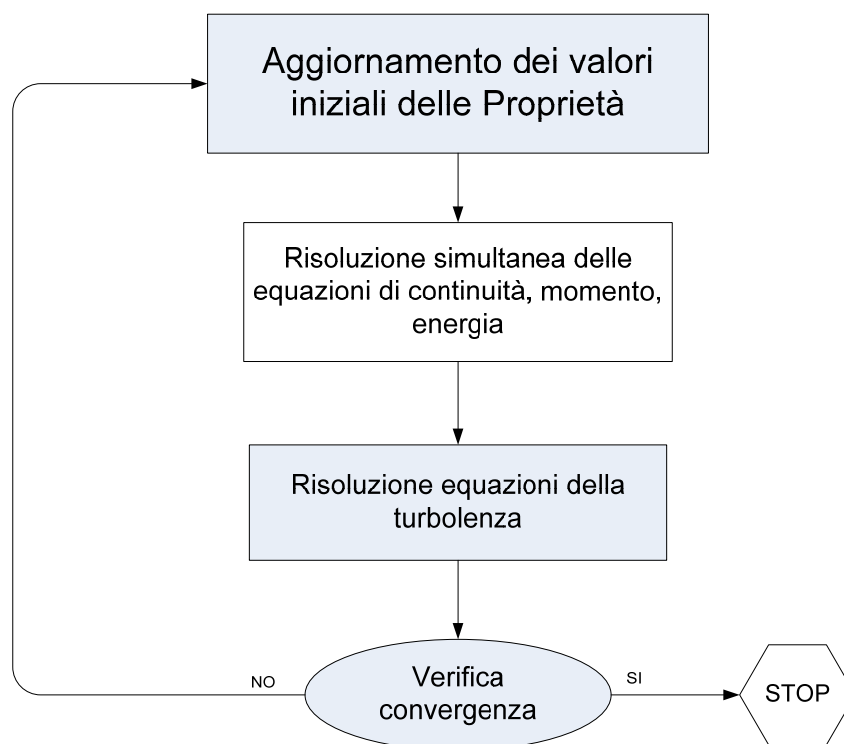


Fig. 3.12 Flusso logico per la convergenza della soluzione

Una volta effettuate le simulazioni i risultati ottenuti sono stati visionati, in primo luogo, per accertarci che il numero di iterazioni impostato fosse sufficiente per portare la soluzione a convergenza. A tale scopo si è creato un grafico su cui vengono visualizzati gli andamenti dei residui e di alcune grandezze (componenti delle velocità, k , ε) in funzione del numero delle iterazioni, accertandoci che andassero a convergenza (vedi Fig. 3.13). Si osservi che il problema della convergenza della soluzione è indipendente dai valori dei parametri di volo. Come si vede dalla Fig. 3.13, relativa ad una simulazione effettuata ad α e β nulli e a 70 m/s, il numero di iterazioni da noi impostato (2500) è sufficiente affinché tutte le grandezze monitorate si portino a dei valori costanti.

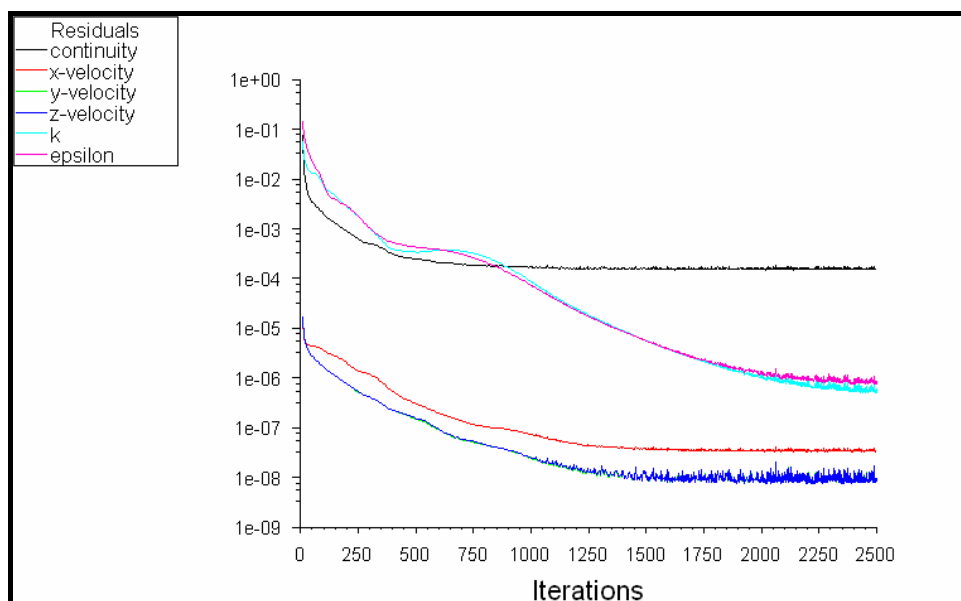


Fig. 3.13 Andamento dei residui

In secondo luogo, è stato verificato che non ci fossero errori nell'impostazione della simulazione stessa. Tali errori infatti porterebbero a dei valori della velocità e delle pressioni sul contorno della sonda non realistici. Quindi, attraverso un display, si è visualizzato l'andamento delle velocità sul piano xy della simulazione effettuata ad α e β nulli e a 70 m/s: queste sono state rappresentate in *Fig. 3.14*, per ogni punto del piano, come vettori individuati da direzione, verso e modulo, di cui si riportano i valori nella scala sulla destra del disegno.

Dalla figura inoltre si possono apprezzare alcune caratteristiche del flusso in esame quali linee di corrente, il punto di ristagno e i vortici che si creano alla base del modello, tenendo presente che tali vortici non sono presenti nella realtà in quanto la sonda ha la forma riportata in *Fig. 1.11*.

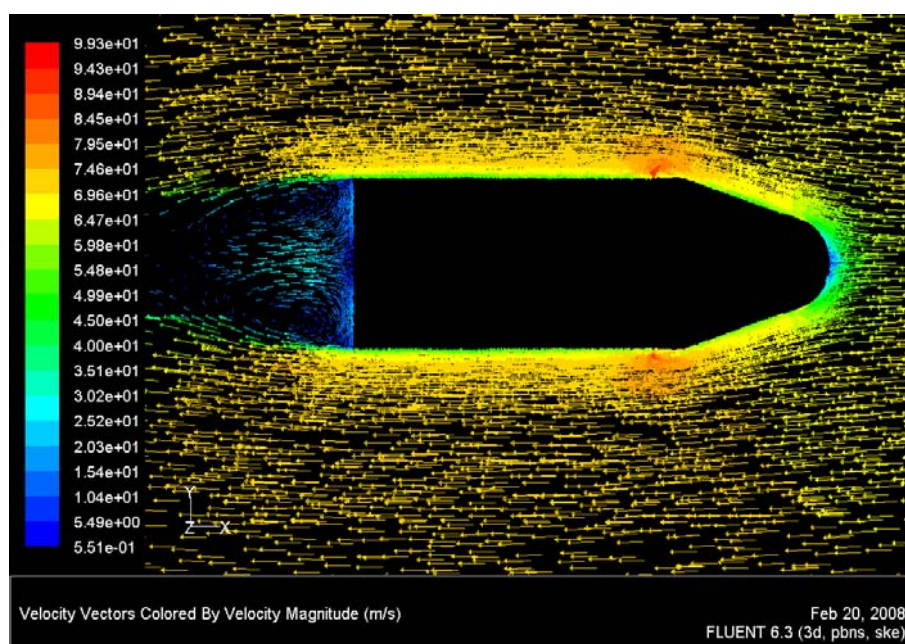


Fig. 3.14 Andamento della velocità sul piano xy

Nelle Fig. 3.15 e Fig. 3.16 sono inoltre visualizzati l'andamento della pressione statica e dinamica sul piano xy osservando che tali risultati sono tutti relativi alla simulazione realizzata ad $\alpha=0^\circ$, $\beta=0^\circ$ ed alla velocità di 70 m/s e quindi sono validi, grazie alla simmetria, per qualunque piano che contenga l'asse del modello.

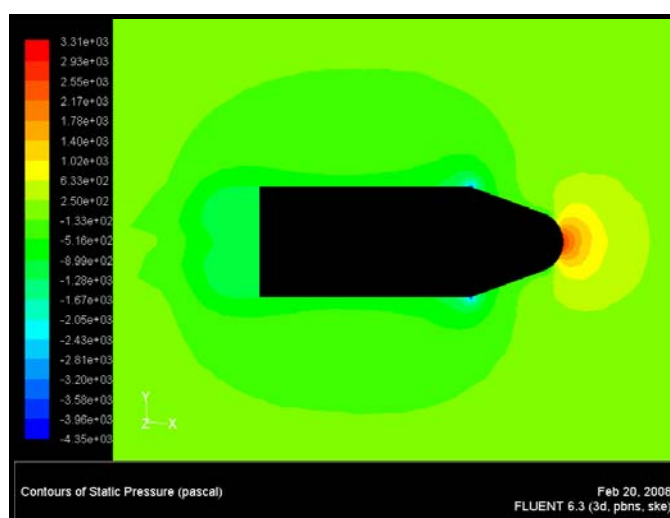


Fig. 3.15 Visualizzazione della pressione statica sul piano xy

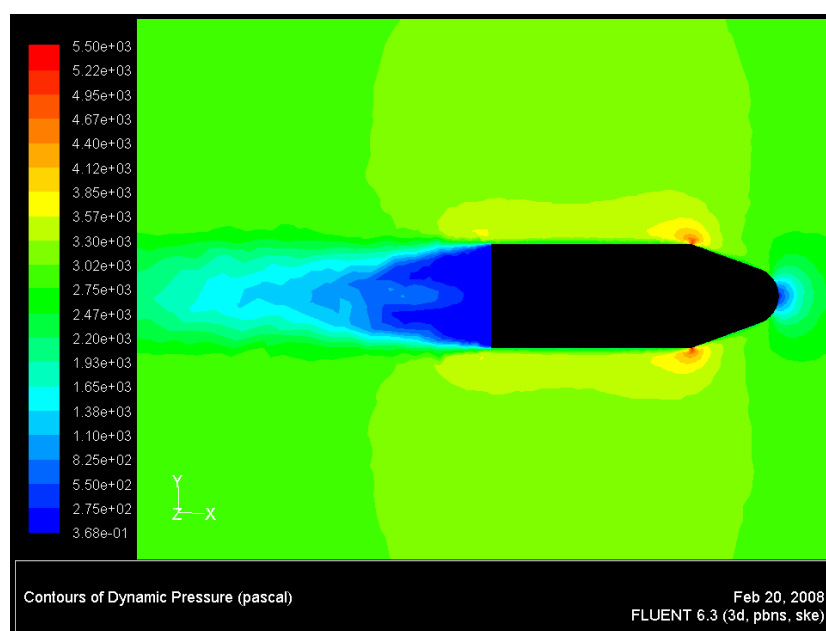


Fig. 3.16 Visualizzazione della pressione dinamica sul piano xy

Successivamente i risultati delle simulazioni sono stati analizzati usando il software *Matlab*[®] per capire se fosse possibile interpolarli diminuendo così il numero delle simulazioni da effettuare. Tale software ci ha permesso di gestire le matrici di dati esportati dall'analisi CFD. Per ogni simulazione sono state esportate delle matrici in cui la prima colonna contiene un numero da 0 a 84 che identifica il punto di controllo preso in considerazione. Sulle altre colonne la matrice presenta le coordinate del punto (x , y , z) ed infine la pressione locale calcolata nel punto. Si osservi che il valore di pressione (ΔP) considerato rappresenta la variazione rispetto alla pressione al *sea level* ed alla temperatura *standard* di 15°C.

Inizialmente sono state valutate le pressioni ai valori estremi dell'intervallo di velocità considerato (20 m/s e 70 m/s): per ognuna delle due velocità si sono presi in considerazione i punti a $\beta=0^\circ$ e gli α positivi (0° , 5° , 10° , 15°). Nelle Fig. 3.17 e Fig. 3.18 sono visualizzati i risultati ottenuti: le variazioni di pressione sono riportate in funzione dei punti di controllo.

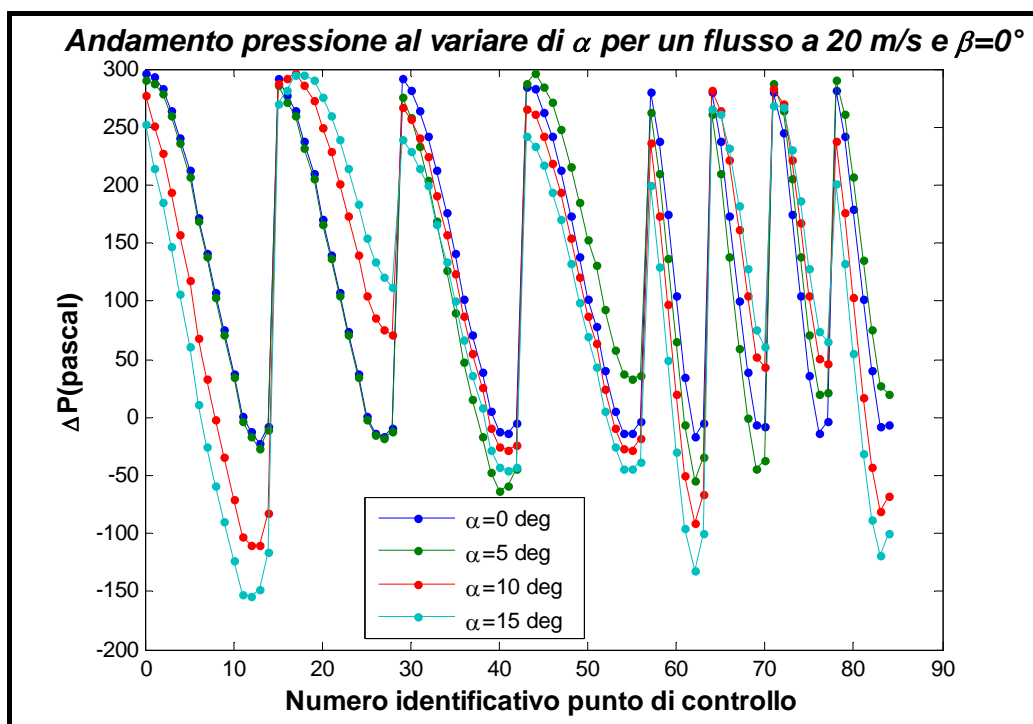


Fig. 3.17 Andamento di ΔP al variare di α con $\beta=0^\circ$ per un flusso a 20 m/s

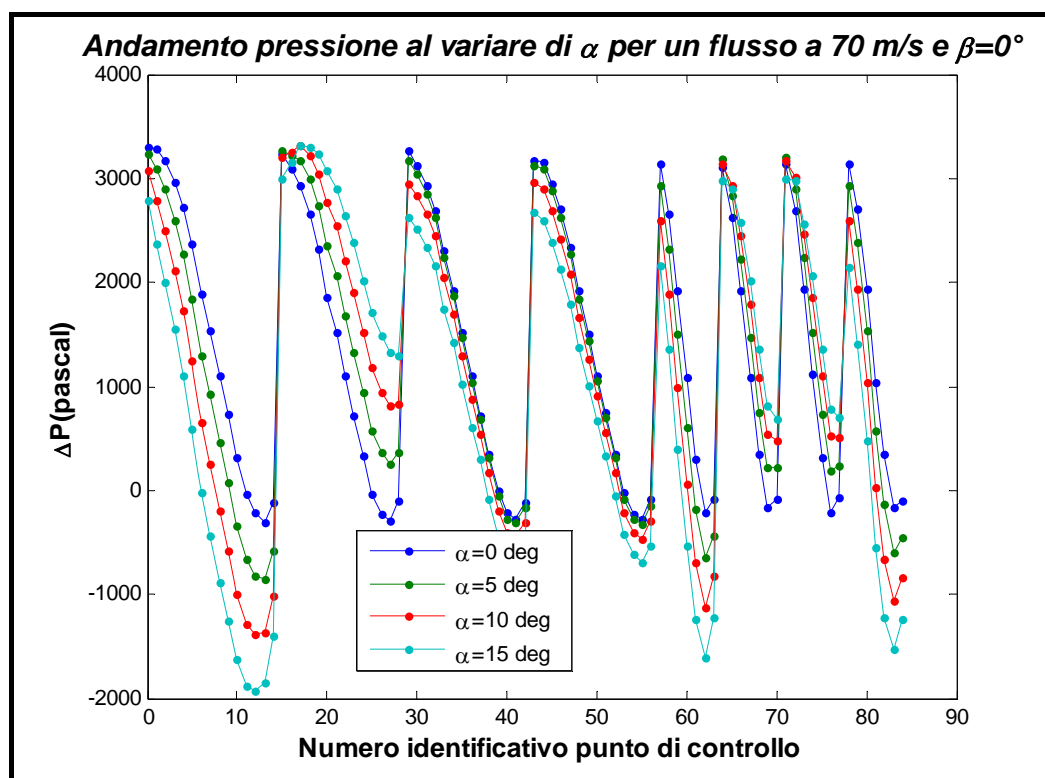


Fig. 3.18 Andamento di ΔP al variare di α con $\beta=0^\circ$ per un flusso a 70 m/s

Da queste simulazioni si è notato che, fissati l'angolo di derapata e la velocità, c'è una relazione tra le pressioni misurate al variare di α , tale da rendere possibile un'interpolazione dei dati. Una relazione simile è stata trovata tra le pressioni misurate ad α e β costanti, facendo variare questa volta la velocità (vedi Fig. 3.19 e Fig. 3.20).

Inoltre si sono riscontrate delle simmetrie nel *database* di pressione dovute alla geometria: fissando α , si può passare da β positivi a β negativi semplicemente facendo un ribaltamento rispetto all'asse z e analogamente fissando β ed eseguendo questa volta il ribaltamento rispetto all'asse y . Nella Fig. 3.21 vengono visualizzati in rosso i dati relativi alla simulazione effettuata ad $\alpha=5^\circ$ e $\beta=-5^\circ$ e relativi al semiarco di circonferenza verticale superiore, in blu quelli relativi ad una simulazione eseguita ad $\alpha=-5^\circ$ e $\beta=10^\circ$ e relativi al semiarco di circonferenza verticale inferiore.

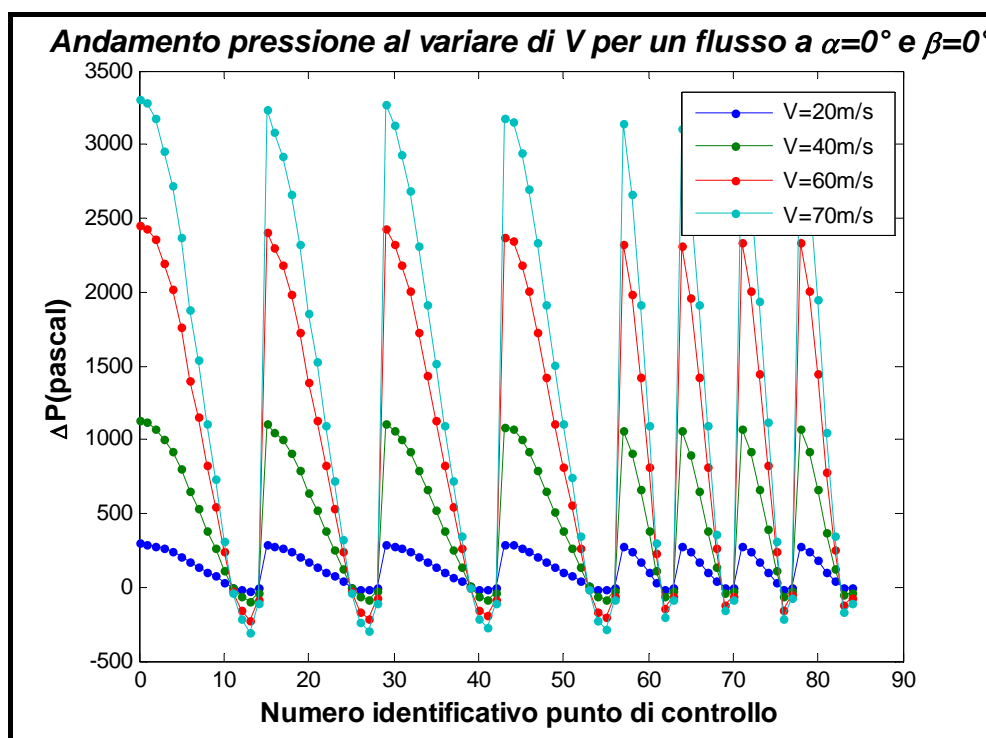


Fig. 3.19 Andamento di ΔP al variare della velocità fissati $\alpha=0^\circ$ e $\beta=0^\circ$

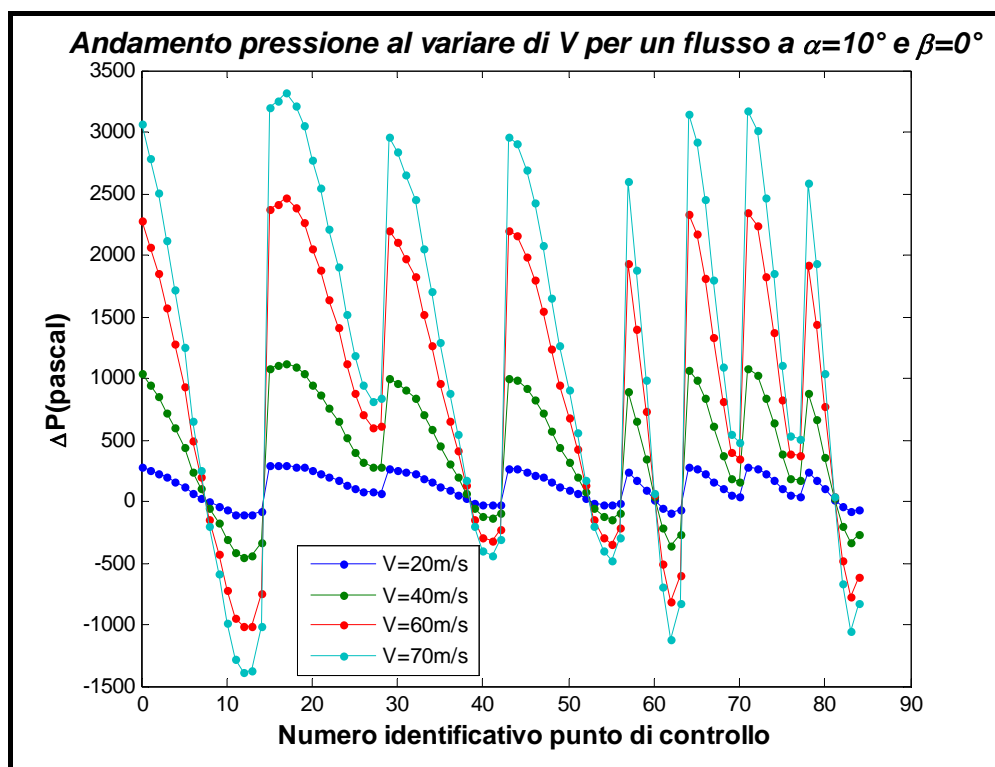


Fig. 3.20 Andamento di ΔP al variare della velocità fissati $\alpha=10^\circ$ e $\beta=0^\circ$

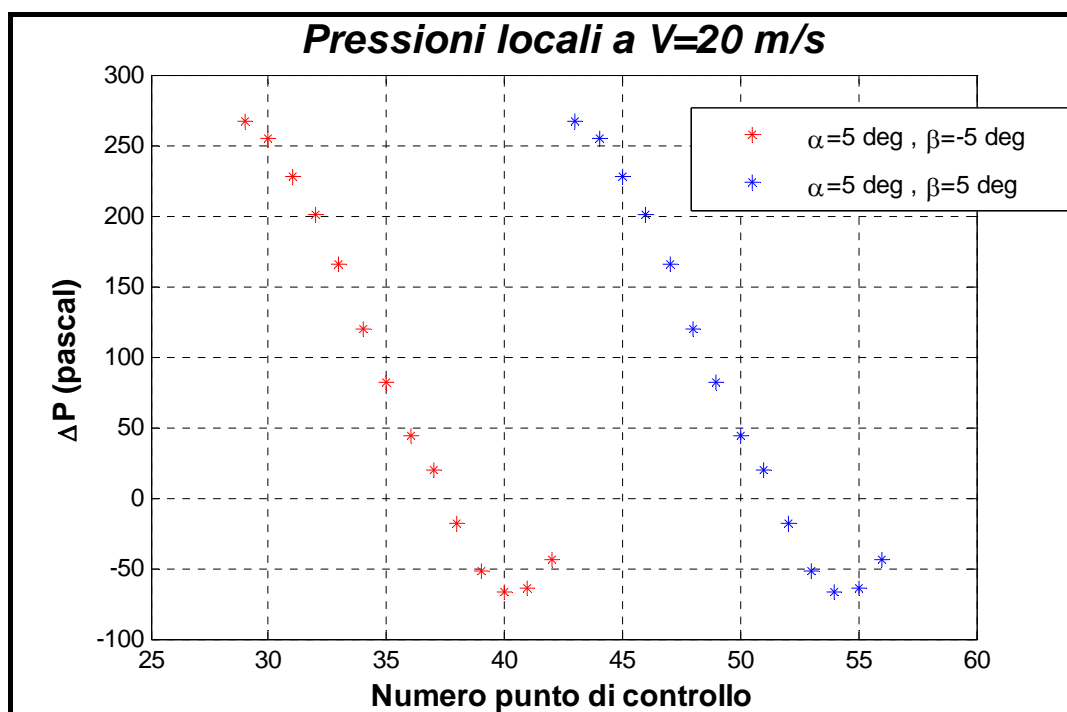


Fig. 3.21 Visualizzazione simmetrie geometriche

Una volta effettuate alcune simulazioni si è cercato sfruttando le proprietà descritte in precedenza di completare la griglia delle simulazioni alle velocità estreme di 20 m/s e 70 m/s utilizzando i dati ricavati attraverso le relazioni di simmetria e le interpolazioni.

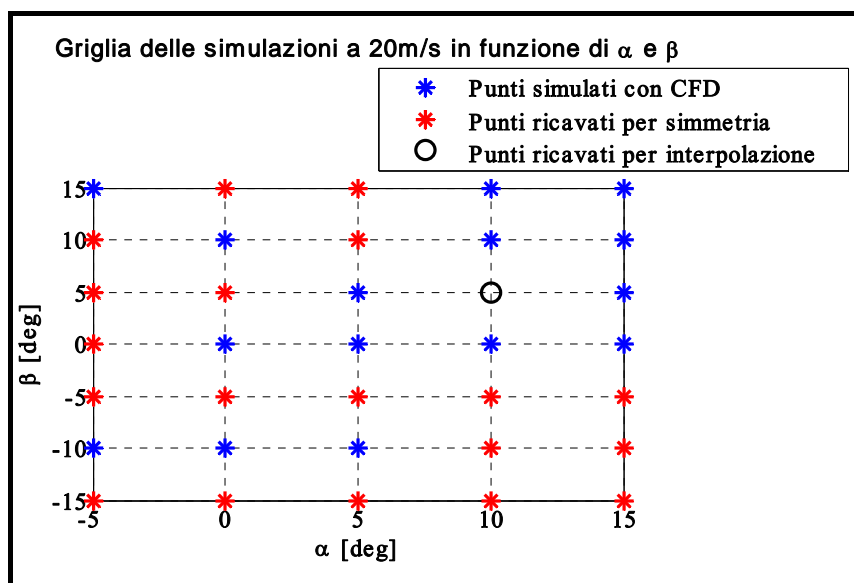


Fig. 3.22 Griglia simulazioni a 20m/s in funzione di α e β

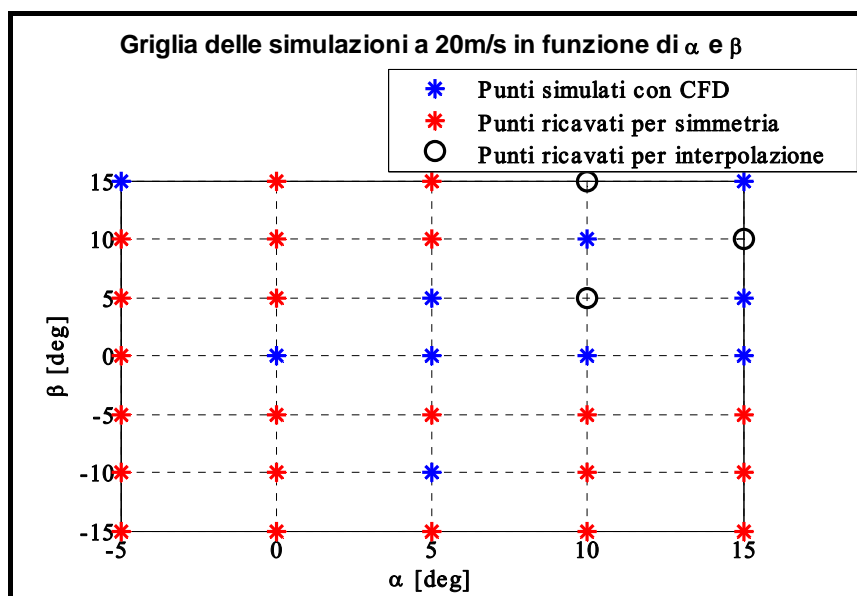


Fig. 3.23 Griglia simulazioni a 70m/s in funzione di α e β

In Fig. 3.22 e Fig. 3.23 sono raffigurate le due griglie delle simulazioni in cui in rosso sono marcati i punti in cui la simulazione è stata realmente eseguita attraverso *Fluent*[®], in blu sono rappresentate le simulazioni ricavate attraverso le simmetrie e in nero sono segnalate le simulazioni interpolate. In Tab. 3.1 sono elencate le simulazioni che sono state effettivamente eseguite con *Fluent*[®], le simulazioni che sono state ricavate per interpolazione e quelle ricavate per simmetrie geometriche. Una volta completate le due griglie a 20 m/s e 70 m/s sono state effettuate tutte le simulazioni ad $\alpha=0$ deg e $\beta=0$ deg con intervalli della velocità di 5 m/s, per avere una stima della variazione di pressione al variare della velocità fissati l'angolo d'incidenza e di derapata (Fig. 3.24).

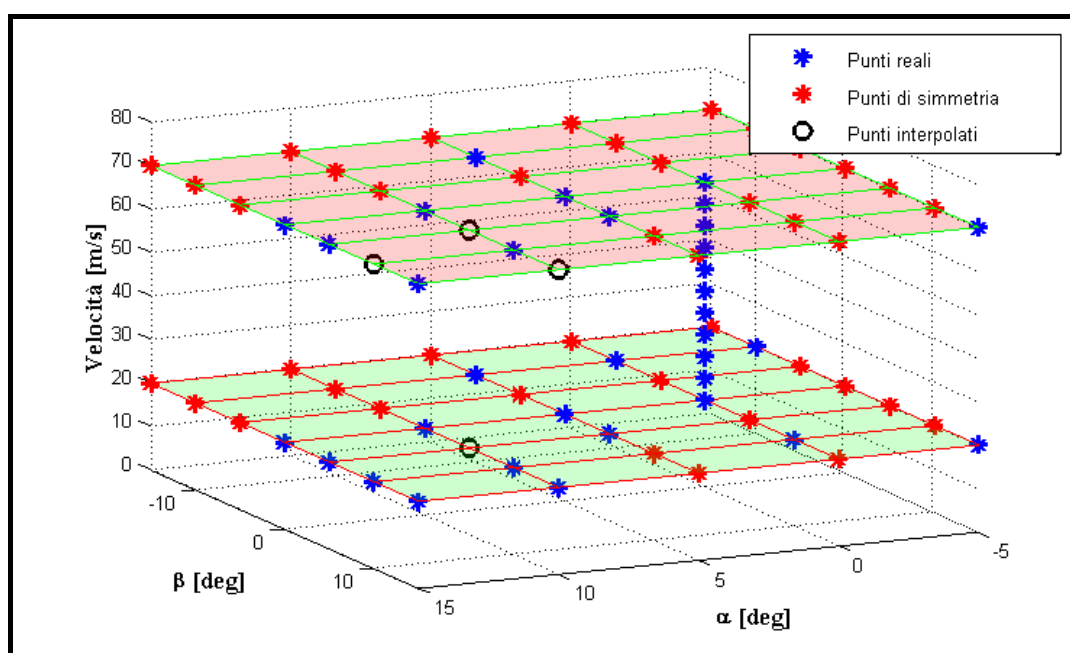


Fig. 3.24 Andamento ΔP al variare di α e β per

Database di Pressione							
Velocità	α [deg]	β [deg]	Dato ottenuto per	Velocità	α [deg]	β [deg]	Dato ottenuto per
20 m/s	0	0	Simulazione	50 m/s	0	0	Simulazione
	-5	0	Simmetria	55 m/s	0	0	Simulazione
	5	0	Simulazione	60 m/s	0	0	Simulazione
	10	0	Simulazione		10	0	Simulazione
	15	0	Simulazione	65 m/s	10	10	Simulazione
	0	5	Simmetria		0	0	Simulazione
	-5	5	Simmetria	70 m/s	0	0	Simulazione
	5	5	Simulazione		-5	0	Simmetria
	10	5	Interpolazione		5	0	Simulazione
	15	5	Simulazione		10	0	Simulazione
	0	10	Simulazione		15	0	Simulazione
	-5	10	Simmetria		0	5	Simmetria
	5	10	Simmetria		-5	5	Simmetria
	10	10	Simulazione		5	5	Simulazione
	15	10	Simulazione		10	5	Interpolazione
	0	15	Simmetria		15	5	Simulazione
	-5	15	Simulazione		0	10	Simmetria
	5	15	Simmetria		-5	10	Simmetria
	10	15	Simulazione		5	10	Simmetria
	15	15	Simulazione		10	10	Simulazione
	0	-5	Simmetria		15	10	Interpolazione
	-5	-5	Simmetria		0	15	Simmetria
	5	-5	Simmetria		-5	15	Simulazione
	10	-5	Simmetria		5	15	Simmetria
	15	-5	Simmetria		10	15	Interpolazione
	0	-10	Simulazione		15	15	Simulazione
	-5	-10	Simulazione		0	-5	Simmetria
	5	-10	Simulazione		-5	-5	Simmetria
	10	-10	Simmetria		5	-5	Simmetria
	15	-10	Simmetria		10	-5	Simmetria
	0	-15	Simmetria		15	-5	Simmetria
	-5	-15	Simmetria		0	-10	Simmetria
	5	-15	Simmetria		-5	-10	Simmetria
	10	-15	Simmetria		5	-10	Simulazione
	15	-15	Simmetria		10	-10	Simmetria
25 m/s	0	0	Simulazione		15	-10	Simmetria
30 m/s	0	0	Simulazione		0	-15	Simmetria
35 m/s	0	0	Simulazione		-5	-15	Simmetria
40 m/s	0	0	Simulazione		5	-15	Simmetria
	10	0	Simulazione		10	-15	Simmetria
45 m/s	0	0	Simulazione		15	-15	Simmetria
50 m/s	15	15	Simulazione				

Tab. 3.1 Tabella riassuntiva del database preliminare

Noto questo legame tra pressione e velocità si è proceduto all'interpolazione dei dati utilizzando, dei pesi calcolati attraverso i dati provenienti dalle simulazioni ad

angoli d'incidenza e di derapata nulli del punto di controllo numero 0 che non è altro che il punto centrale sulla calotta sferica della sonda. Questo peso viene applicato al generico punto di controllo e ai generici α e β attraverso la relazione:

$$\Delta P(Vel, \alpha, \beta, N) = (f_p \cdot diff) + \Delta P(20 \frac{m}{s}, \alpha, \beta, N) \quad (Eq. 3.4)$$

Con

$$f_p = \frac{\left(\Delta P(Vel, 0, 0, 0) - \Delta P(20 \frac{m}{s}, 0, 0, 0) \right)}{\left(\Delta P(70 \frac{m}{s}, 0, 0, 0) - \Delta P(20 \frac{m}{s}, 0, 0, 0) \right)} \quad (Eq. 3.5)$$

e

$$diff = \Delta P(70, \alpha, \beta, N) - \Delta P(20, \alpha, \beta, N) \quad (Eq. 3.6)$$

In cui le grandezze hanno il seguente significato: ΔP è una funzione che fornisce il valore del ΔP in funzione della velocità Vel , degli angoli di incidenza α e di derapata β e del numero del punto di controllo preso in considerazione N . Inoltre sono state fatte delle simulazioni di controllo elencate in *Tabella 3.1* che permettessero di fare una stima dell'errore che si commette nel procedere a questa operazione, vedi *Fig. 3.25* in cui è rappresentato l'errore commesso durante le simulazioni nei punti in cui sono state fatte le simulazioni di controllo. Da questo grafico si vede come gli errori siano molto piccoli, dell'ordine di 15 Pa, che su misure dell'ordine di migliaia di Pa sono trascurabili.

SIMULAZIONI DI CONTROLLO			
VELOCITA'	ALFA	BETA	Caratteristica
40 m/s	10	0	Simulazione di controllo
50 m/s	15	15	Simulazione di controllo
60 m/s	10	0	Simulazione di controllo
60 m/s	10	10	Simulazione di controllo

Tabella 3.1 Elenco simulazioni di controllo

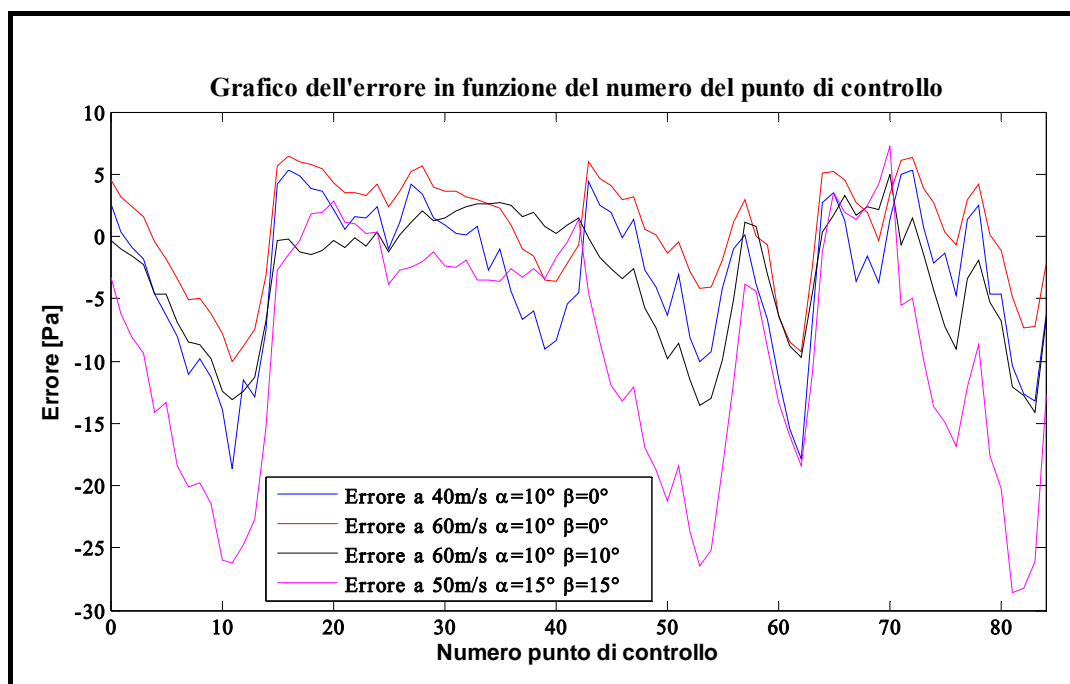


Fig. 3.25 Grafico errore commesso durante le interpolazioni

Verificato che gli errori commessi in queste interpolazioni fossero piccoli si è completato l'intero inviluppo di volo generando una griglia di 385 punti.

Per ogni punto di controllo si è creata una matrice di dati tridimensionale capace di, attraverso l'inserimento degli angoli di incidenza, di derapata e della velocità, fornire il valore di ΔP nel punto considerato. Nella Fig. 3.26 è visualizzato l'andamento delle ΔP al variare di α e β nel punto di controllo indicato dal numero 0, alla velocità di 20 m/s.

E' stata costruita una superficie analoga a quella mostrata in Fig. 3.26, per ogni valore delle velocità in esame e per ogni punto di controllo.

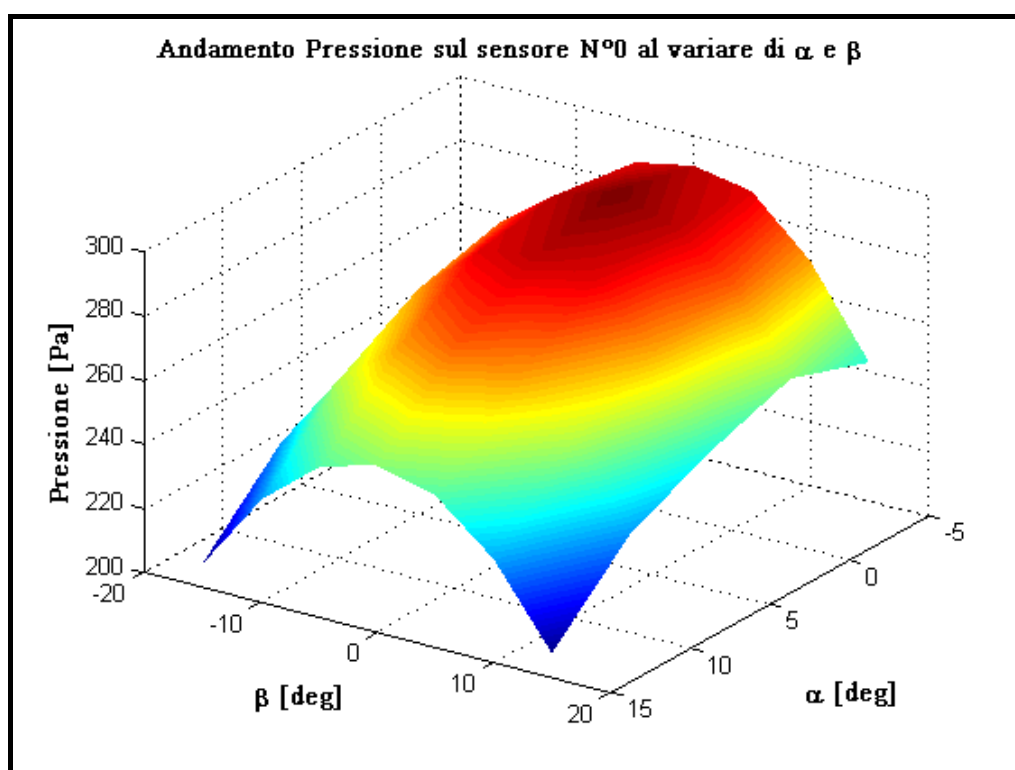


Fig. 3.26 Andamento ΔP al variare di α e β per un flusso a 20 m/s

PARTE II: *Ricostruzione dei parametri di volo
mediante reti neurali*

4 Le Reti Neurali

4.1 Campi d'applicazione e possibili problematiche

Una rete neurale artificiale, *Neural Network* (NN), è un sistema di elaborazione dell'informazione che ha le caratteristiche comuni alle reti neurali biologiche ed è costituita da semplici unità interconnesse fra loro in modo più o meno complesso.

Le reti neurali artificiali nascono nel 1943 ad opera di *McCulloch* e *Pitts* che propongono un modello di neurone artificiale e la connessione di un gran numero di queste unità in rete. Nel 1962 nasce *Perceptron* la prima macchina in grado, attraverso un meccanismo di apprendimento basato sulla determinazione in linea dei pesi sintattici, di apprendere ad esempio il funzionamento di molti degli operatori matematici. Ma è soltanto nel 1985 che *Rumelhart*, *Hinton* e *Williams*, attraverso l'introduzione di un potente algoritmo di apprendimento (*backpropagation*), rilanciano lo studio delle reti neurali, facendo sì che oggi siano impiegate per risolvere problemi complessi in diversi campi [13]:

- Sistemi intelligenti basati su ingressi di segnali di origine sensoriale (ad es. visiva, olfattiva, acustica, tattile).
- Sistemi di controllo adattivi (sistemi di integrazione sensoriale-motore, sistemi intelligenti di controllo dei processi, sistemi intelligenti di controllo di apparecchiature).
- Elaborazione adattiva dei segnali (compressione dei dati in un PC, comunicazione e archivio, miglioramento e sintesi dei dati, memorie associative).
- Sistemi esperti.
- Robot autonomi.

Una NN può essere definita come una rete di elementi semplici interconnessi in parallelo in larga misura e la cui organizzazione gerarchica è progettata in modo da interagire con gli oggetti del mondo reale in modo simile al sistema nervoso

centrale. Ogni rete neurale è composta da un numero più o meno elevato di unità elementari, chiamate anche “neuroni artificiali”, “celle” o “nodi”, modelli matematici dei neuroni biologici, che ricevono dei segnali in ingresso e li elaborano creando un unico segnale d’uscita, inviato successivamente ad altri neuroni. Essendo semplicemente un modello matematico, il neurone non cattura appieno le proprietà biologiche e funzionali dei neuroni reali, ma costituisce una forte semplificazione della realtà mostrando tuttavia delle analogie che ne permettono l’utilizzo.

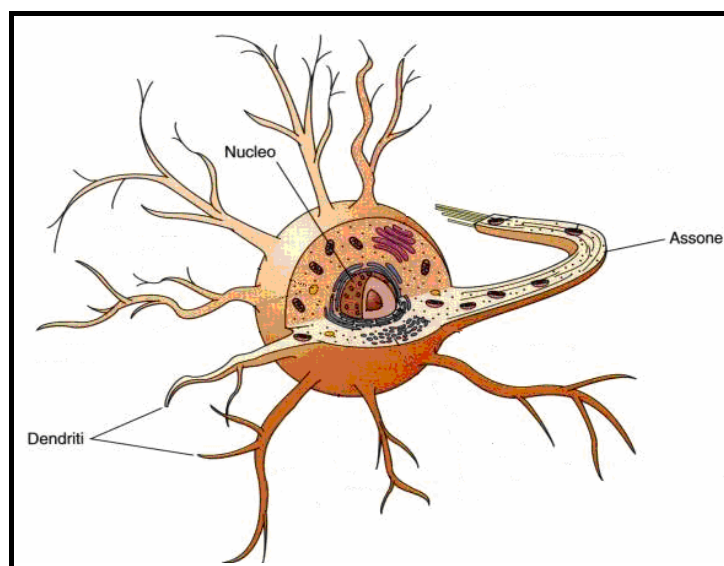


Fig. 4.1 Schema di un neurone biologico

Le strategie più comuni per semplificare le conoscenze sui neuroni biologici e caratterizzare matematicamente delle unità artificiali, che possano potenzialmente simulare i meccanismi di calcolo che avvengono nel cervello umano, sono le seguenti [8]:

- 1) Ignorare le non linearità delle relazioni fra gli *spikes* presinaptici e le attività postsinaptiche.
- 2) Ignorare le non linearità nelle relazioni tra le attività dei canali ed il potenziale postsinaptico.

- 3) Incorporare gli effetti della struttura dendrica nei pesi sinaptici.
- 4) Permettere ai pesi sinaptici di cambiare segno.

La j -esima unità riceve gli *input*, ovvero gli stimoli della rete, da strati precedenti o direttamente dall'esterno. Facendo un paragone tra la terminologia medica e quella neuronale possiamo dire che su ogni connessione sono presenti dei pesi che caratterizzano la “forza” della connessione (corrispondente alla sinapsi umana). Questi segnali sono convogliati attraverso i dendriti, insieme alla soglia interna del neurone (bias), nel *net* (soma) del neurone artificiale. L'uscita del neurone (assone) è generata dalla funzione di trasferimento $f(net)$.

A titolo di esempio, considerando un neurone Y che riceve in ingresso i segnali dai neuroni X_1, X_2, \dots, X_n :

$$Y_{IN} = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n + Y_{BIAS} \quad (Eq. 4.1)$$

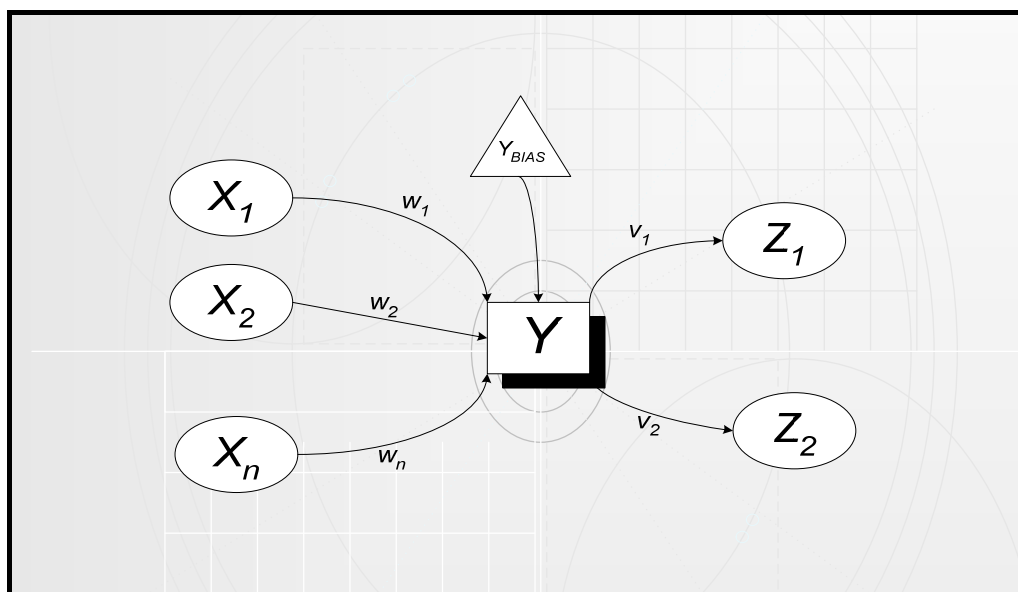


Fig. 4.2 Neurone artificiale: schema di funzionamento

Dove i termini w_1, w_2, \dots, w_n sono i pesi sinaptici e Y_{BIAS} è la soglia d'ingresso del neurone. Ad Y_{IN} viene applicata la funzione di trasferimento ottenendo il segnale d'uscita che verrà inviato a tutti i neuroni dello strato successivo. Nella

rete c'è completa connessione tra i neuroni in quanto nella progettazione di una rete neurale occorre considerare una serie di problemi legati principalmente alla sua architettura, al *database* usato per l'apprendimento e all'algoritmo di apprendimento stesso. Per quanto riguarda l'architettura, i neuroni vengono organizzati in strati o *layer*, si parla dunque di reti a singolo strato, multistrato, se sono presenti più strati interni, detti *hidden layer*: in questa situazione i neuroni d'ingresso non sono considerati un vero e proprio strato perché non eseguono alcuna elaborazione a differenza dello strato formato dai neuroni di *output*. Ogni neurone di uno strato è collegato a tutti quelli dello strato successivo e collegamento è caratterizzato dal proprio peso sinaptico. Il numero di neuroni per ogni strato, pesi, soglie d'ingresso, funzione di trasferimento costituiscono l'architettura di una rete ed indicano la capacità di adattarsi a svolgere i compiti assegnati. Esistono in letteratura delle metodologie, chiamate di *pruning* o *surgering*, che permettono di migliorare le prestazioni della rete ad un costo computazionale minore eliminando alcune delle connessioni all'interno della rete.

Altro aspetto importante per il corretto funzionamento di una rete è la fase di apprendimento, durante la quale vengono decisi i valori dei pesi. Infatti bisogna distinguere tra il semplice immagazzinamento degli esempi e l'acquisizione delle regole astratte che generano gli esempi. Naturalmente il corretto funzionamento di una rete neurale sull'insieme di apprendimento non offre garanzia di un altrettanto corretto funzionamento su altri dati relativi allo stesso concetto ma non appartenenti al *database* di *training*. Infatti quando il numero delle unità cresce, aumenta in corrispondenza anche il potere computazionale, ma la capacità di generalizzare su nuovi esempi diminuisce dato che il *fitting* sull'insieme di *training* ha luogo in uno spazio di parametri enorme, ma vincolati da pochi esempi.

Esistono due tipi principali di apprendimento:

- 1) *Supervised learning*
- 2) *Unsupervised learning*

Nella modalità di apprendimento con supervisione, l'addestramento è basato sulla disponibilità di un insieme di coppie di dati:

- dati del problema.
- corrispondenti soluzioni (*target*)

Ad ogni vettore d'ingresso corrisponde un vettore d'uscita che rappresenta il comportamento voluto dalla rete. Una volta fornite queste informazioni, i pesi della rete si calcolano con modalità diverse a seconda dell'algoritmo di addestramento utilizzato [5].

L'apprendimento senza supervisione non fa riferimento ad una casistica precostituita di esempi, ma al contrario viene fornito il solo insieme degli ingressi e la rete modifica automaticamente i pesi, in modo che vettori d'ingresso simili attivino le stesse unità d'uscita (*cluster*). Questo tipo di apprendimento è utilizzato per addestrare le reti *Kohonen* o *Self Organizing Map* (SOM).

4.2 Reti neurali statiche e metodologie di addestramento

Le reti neurali statiche descrivono il comportamento di sistemi statici, ovvero il vettore d'uscita, in un determinato istante, dipende unicamente dal valore in quell'istante del vettore d'ingresso.

L'architettura più comune di una rete statica è il perceptrone multi-strato (*Multi-Layer Perceptron*, MLP), che presenta uno strato d'ingresso, uno o più strati intermedi e uno strato d'uscita, come mostrato in *Fig. 4.3*.

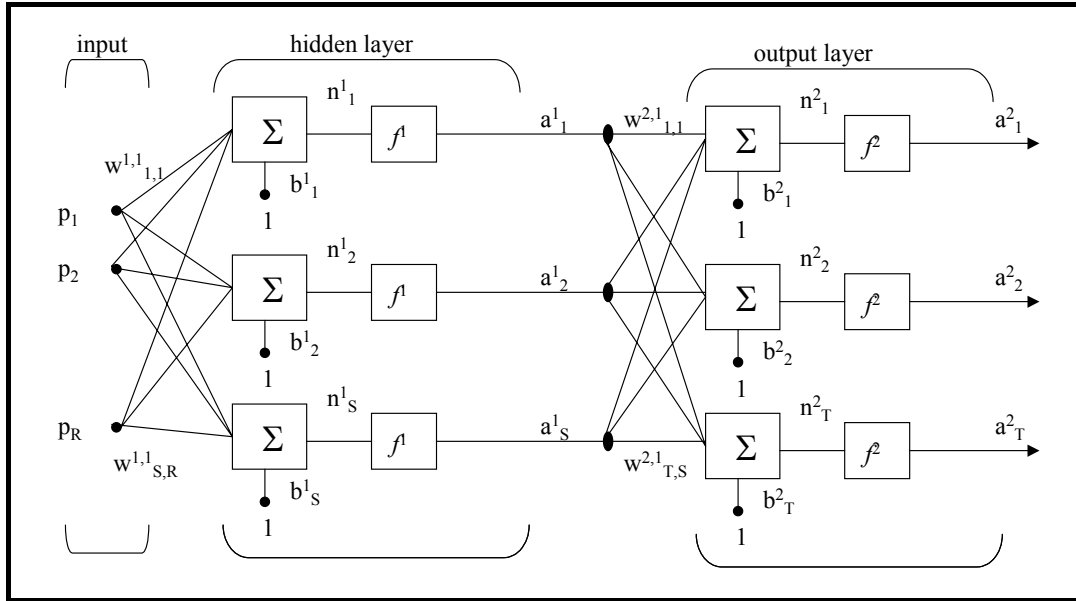


Fig. 4.3 Multi-Layer Perceptron ad un solo strato nascosto

Il vettore degli ingressi p è costituito da un numero R di elementi e ad ogni strato di neuroni nascosto verrà associata una matrice, costituita dai pesi delle connessioni, di dimensioni $R \times S$, con S il numero di neuroni dello strato. Questa matrice è costituita dagli elementi w_{ij} che rappresentano i pesi della connessione fra il neurone i -esimo dello strato nascosto e l'ingresso j -esimo. Ogni strato, inoltre presenta un vettore b , di lunghezza S , costituito dalle soglie di attivazione dei neuroni formanti quello strato.

L'addestramento è un procedimento iterativo in cui i valori dei pesi e delle soglie di attivazione vengono riaggiornati, passo dopo passo, fino a quando una funzione differenza degli errori fra l'uscita della rete e l'uscita del sistema reale è minore di un certo valore (*goal*). Tale funzione, anche chiamata cifra di merito C , è espressa da:

$$C(E_i) = C(a_i(p_i) - a_i^{NN}(p_i, \Theta)) \quad (\text{Eq. 4.2})$$

con

- p_i : ingresso i -esimo

- a_i : risposta del sistema reale all' i -esimo ingresso
- a_i^{NN} : risposta della rete all' i -esimo ingresso
- Θ : vettore dei parametri della rete neurale

La cifra di merito è dunque elaborabile come somma degli errori quadratici (*Sum of Squared Error*, SSE), come somma normalizzata degli errori quadratici (*Normalized Sum of Squared Error*, NSSE) oppure come media degli errori quadratici (*Mean of Squared Error*, MSE).

L'obiettivo dell'addestramento è trovare i parametri della rete che minimizzano il valore della cifra di merito. Il vettore Θ viene quindi riaggiornato per ogni iterazione durante l'addestramento attraverso la relazione:

$$\Theta^{(i+1)} = \Theta^{(i)} + \mu^{(i)} \cdot f^{(i)} \quad (Eq. 4.3)$$

dove con i è indicata i -esima iterazione o epoca, $f^{(i)}$ è il parametro di direzione di ricerca e $\mu^{(i)}$ è il passo (*learning rate*) dell'algoritmo iterativo. Durante l'addestramento di una rete viene, quindi, cercato il punto di minimo della superficie che rappresenta l'andamento dell'errore nello spazio dei pesi, eseguendo la ricerca nella direzione dell'opposto del gradiente, valutato alla i -esima epoca di addestramento.

Questa forma di ricerca dell'errore usata negli algoritmi di addestramento viene chiamata discesa del gradiente. Per reti ad un solo strato, che utilizzano questo algoritmo, l'aggiornamento dei pesi e del gradiente è immediato: ciò non accade nelle reti multi-strato in cui l'aggiornamento del gradiente implica una procedura di propagazione all'indietro dell'errore, dato che la cifra di merito è calcolata solo con le uscite della rete. In questo caso si parla di algoritmi di addestramento *backpropagation*. Dal punto di vista metodologico si procede in due fasi, nella prima si procede dagli ingressi verso le uscite calcolando le attivazioni di tutti i neuroni; nella seconda, al contrario, si procede a ritroso calcolando di quanto debbano essere modificati i pesi uno per uno. Un inconveniente a cui si può andare

incontro nella *backpropagation* è quello di avere delle superfici dell'errore con minimi locali che inevitabilmente bloccherebbero l'addestramento impedendo alla rete la ricerca del minimo assoluto. Per risolvere questo problema può essere utilizzata una regola di *learning* che cambia la direzione di ricerca.

Uno degli algoritmi principali di addestramento è l'algoritmo di *Newton* che utilizza come matrice per cambiare la direzione di ricerca l'inversa dell'*hessiano* della cifra di merito, vedi Fig. 4.4.

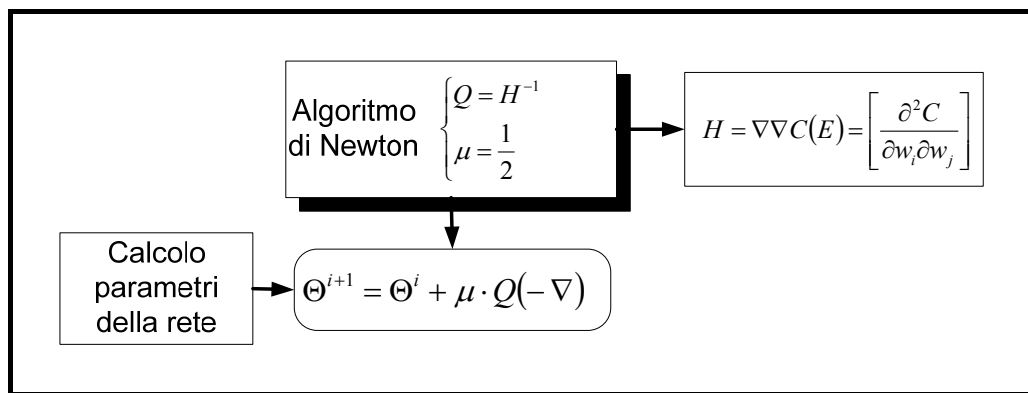


Fig. 4.4 Algoritmo di Newton

In alcuni casi la matrice H è singolare , quindi non invertibile, e viene allora scomposta nel prodotto di tre matrici:

$$H = U \cdot \Sigma \cdot V^T \quad (Eq. 4.4)$$

dove Σ è la matrice diagonale degli autovalori di H . Invece di calcolare l'inversa di H si calcola l'inversa della matrice

$$H + \lambda \cdot I = U \cdot [\Sigma + \lambda \cdot I] \cdot V^T \quad (Eq. 4.5)$$

Questo metodo è chiamato metodo di *Levenberg-Marquardt* ed uno dei metodi di addestramento più usati nonostante sia meno preciso e veloce del metodo di *Newton*.

4.2.1 Metodo di addestramento Early Stopping

La capacità di generalizzare è la caratteristica principale che una rete deve possedere, e si è verificato che, aumentando il numero di epoche di addestramento, l'errore sul *training set* diminuisce; ma questo non vuol dire che la rete possiede una buona capacità di generalizzazione che deve, quindi, essere misurata su un insieme di dati di validazione.

La complessità, intesa come numero di neuroni e numero di *layer*, di una rete svolge, in questo problema, un ruolo importante. Aumentando il numero dei parametri aumentano i gradi di libertà del modello, cioè significa che è possibile modificare i parametri in modo da far aderire maggiormente il comportamento del modello a quello del sistema. Questo però accade fino ad un certo valore della complessità, in cui gli errori sul *training set* e sul *validation set* diminuiscono di pari passo, dopodiché l'errore sul *validation set* inizia ad aumentare nonostante la contemporanea diminuzione dell'errore sul *training set*.

Tale fenomeno prende il nome di *overfitting* o *overlearning* e si verifica dal momento in cui il modello scelto aderisce troppo ai dati di addestramento.

Un metodo per evitare questo problema è l'algoritmo dell'*Early Stopping* (ES). Con questo metodo l'addestramento viene arrestato quando si arriva in prossimità del minimo dell'errore su un *set* di validazione. Ai dati di addestramento vengono affiancati i dati di validazione che la rete usa contemporaneamente all'addestramento per testare la rete ed avere una stima dell'errore che questa commetterebbe su dati diversi da quelli del *training set*. Un aspetto negativo legato all'utilizzo di questo metodo è la scelta opportuna del numero dei dati di *training* e validazione.

4.2.2 Metodo di addestramento Bayesian Regularization

Il metodo *Bayesian Regularization* (BR) migliora la generalizzazione delle reti durante l'addestramento. La cifra di merito, che viene minimizzata per aggiornare i pesi delle connessioni, è pesata attraverso un parametro γ :

$$C = \gamma \cdot MSE + (1 - \gamma) \cdot MSW \quad (4.6)$$

con MSW media dei pesi quadratici (*Mean Squared Weight, MSW*).

Il metodo *Bayesian Regularization* tratta l'addestramento delle reti e la generalizzazione in modo statistico. Infatti, poiché solitamente nel processo di ottimizzazione, l'inizializzazione dei pesi è del tutto casuale (ovvero non influenza i risultati dell'addestramento), in questo metodo di addestramento le soluzioni vengono combinate insieme per trovare quella che produce la generalizzazione migliore. Per esempio, dato un *set* di *training* D ed una rete addestrata con tale *set* di dati, viene calcolato quello che sia il più probabile vettore di *output* y^* per un vettore d'*input* x non appartenente a D .

$$y^* = \int y(x, w) \cdot P(w|D) \cdot dw \quad (4.7)$$

Ogni possibile uscita $y(x, w)$ associata ad un set di pesi w viene pesata con la probabilità posteriore, ovvero la probabilità che D dia il *set* di pesi w , data dal teorema di *Bayes*:

$$P(w|D) \propto P(D|w) \cdot P(w) \quad (4.8)$$

dove

$$P(w|D) = \exp(-MSE) \quad (4.9)$$

è la probabilità che ogni *set* di pesi w dia i *target* di D ed ha una distribuzione di *Gauss* con deviazione *standard* $\gamma^{-\frac{1}{2}}$.

Invece

$$P(w) = \exp(-MSW) \quad (4.10)$$

chiamata probabilità anteriore, è la probabilità dei differenti gruppi di pesi ed ha una distribuzione di *Gauss* con deviazione standard $(1 - \gamma)^{-\frac{1}{2}}$.

Il parametro di regolarizzazione γ è calcolato ed ottimizzato durante l'addestramento. Calcolato l'integrale (4.7) per un certo numero di *set* di pesi e date le (4.8), (4.9) e (4.10) il *set* di pesi che generalizza meglio la rete sarà quello con la cifra di merito più piccola. Il metodo *Bayesian Regularization*, rispetto al metodo *Early Stopping*, richiede un grosso sforzo computazionale ma di solito fornisce risultati migliori.

5 *Individuazione dei parametri principali delle reti*

Il *database* di pressioni, proveniente dall'analisi CFD, è stato ampliato e suddiviso in tre *set* per andare a costituire gli insiemi di addestramento, validazione e test. L'individuazione dei parametri principali delle reti si è svolta analizzando tutte le possibili architetture delle reti e cercando di valutare al meglio quelli che potevano essere i parametri in gioco per un corretto funzionamento sia nella fase di addestramento che di test. Si è partiti dalla caratterizzazione, attraverso i dati provenienti dall'analisi CFD, del *database* di pressioni, che in questa fase verrà ampliato e successivamente diviso in tre *database* che serviranno per le fasi di addestramento, *test* e validazione. Come già detto precedentemente, l'analisi CFD ha fornito un *set* di dati che sono stati inseriti in una matrice di 385 punti a tre dimensioni, in cui ogni punto viene caratterizzato con le coordinate α , β e velocità. Per un addestramento accurato è stato necessario ampliare questo *set* di dati fino a creare un *database* di 10000 punti dei quali solo 1010 sono stati utilizzati per il *training*: gli altri punti sono stati divisi in due gruppi, il primo, costituito da 8450 punti, per eseguire i *test* sulle reti già addestrate e valutarne le prestazioni, il secondo, costituito dai restanti 540 punti, per eseguire una validazione durante l'addestramento.

Per ampliare il *database* aerodinamico si utilizza una *lookup table* (3-D), blocco di *Simulink/Matlab*[®], che serve per ricavare tramite interpolazione i valori di una funzione in tre variabili, che nel nostro caso è il valore delle pressioni in funzione dei parametri di volo. Partendo dai dati inseriti dall'esterno, nel nostro caso il *database* aerodinamico proveniente dall'analisi CFD, vengono generati in modo *random* α , β e velocità; se i dati generati sono mancanti allora viene eseguita un'interpolazione delle pressioni che corrispondono a quella terna di valori. Il modello *Simulink* in Fig. 5.1 mostra i blocchi che generano casualmente i valori di α , β e velocità nel punto in cui si vuole ricavare la pressione.

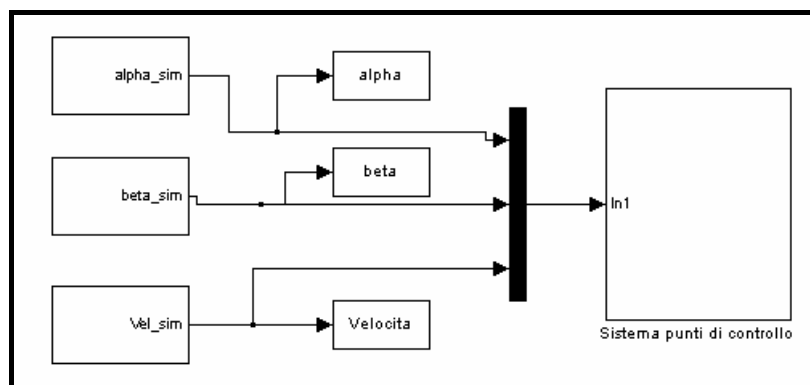


Fig. 5.1 Modello Simulink per l'ampliamento del database preliminare

Nella generazione di valori *random* viene inserito un blocco di saturazione in modo da evitare che il valore generato cada al di fuori dell'involuppo di volo. Tale blocco filtra il valore e lo limita all'interno dell'intervallo preso in considerazione. Per l'angolo d'incidenza i valori vengono limitati tra -5° e 15° , per l'angolo di derapata tra -15° e 15° e della velocità tra 20 e 70 m/s. In Fig. 5.2 sono visualizzati rispettivamente il blocco di generazione *random* dell'*input*, il blocco di saturazione e l'uscita.

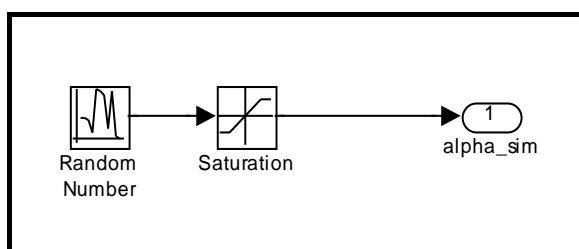


Fig. 5.2 Limitazione dell'ingresso tramite blocco di saturazione

Come si vede in Fig. 5.1 i tre ingressi vengono inviati ad un unico blocco all'interno del quale sono presenti le *lookup tables*, una per ogni punto di controllo considerato.

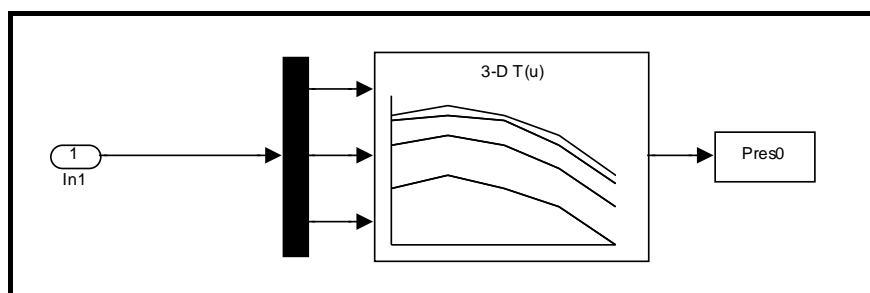


Fig. 5.3 Lookup table

Ogni *lookup table* riceve i tre *input* (α , β e velocità) (Fig. 5.3) ed interpola il valore di pressione nei punti intermedi generati in modo *random*. Il modello Simulink sviluppato presenta 85 *lookup tables* contenute nel blocco “Sistema punti di controllo” di Fig. 5.1, una per ogni sensore. In Fig. 5.4, viene mostrata la struttura interna della *lookup table* relativa al sensore posto al centro della parte anteriore della sonda, denominato con il numero 0.

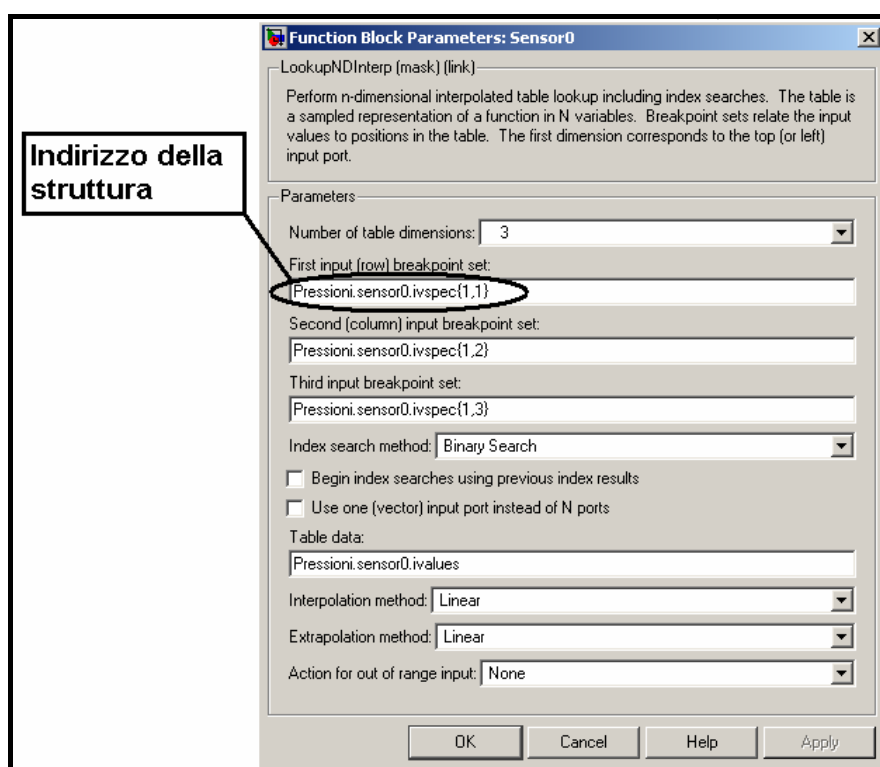


Fig. 5.4 Menù d'impostazione della lookup table

Per impostare la *lookup table* si inserisce l'indirizzo della struttura da cui poter prendere i dati d'ingresso per eseguire l'interpolazione, vedi Fig. 5.4. Alla fine del processo per ogni punto di controllo viene creato un vettore: su ogni riga, che corrisponde dunque ad un punto dell'involuppo, è esportata la terna di valori α , β e velocità, ed il valore di pressione generato dalla *lookup table*.

Il *database* così ampliato è stato suddiviso in tre gruppi, tenendo conto di diversi fattori. Per quanto riguarda il *training set* occorre scegliere un numero sufficientemente grande di punti ed inoltre questi punti devono presentare una distribuzione omogenea, che contenga anche i punti di bordo dell'involuppo in modo da ottimizzare la fase di addestramento stesso. A tale scopo è stata sviluppata una *routine* di *Matlab*® in cui tramite il comando “convhulln” è possibile creare un guscio al *database* (vedi Fig. 5.5): i punti del *database* appartenenti al guscio vengono inseriti nel *training set* come punti di bordo.

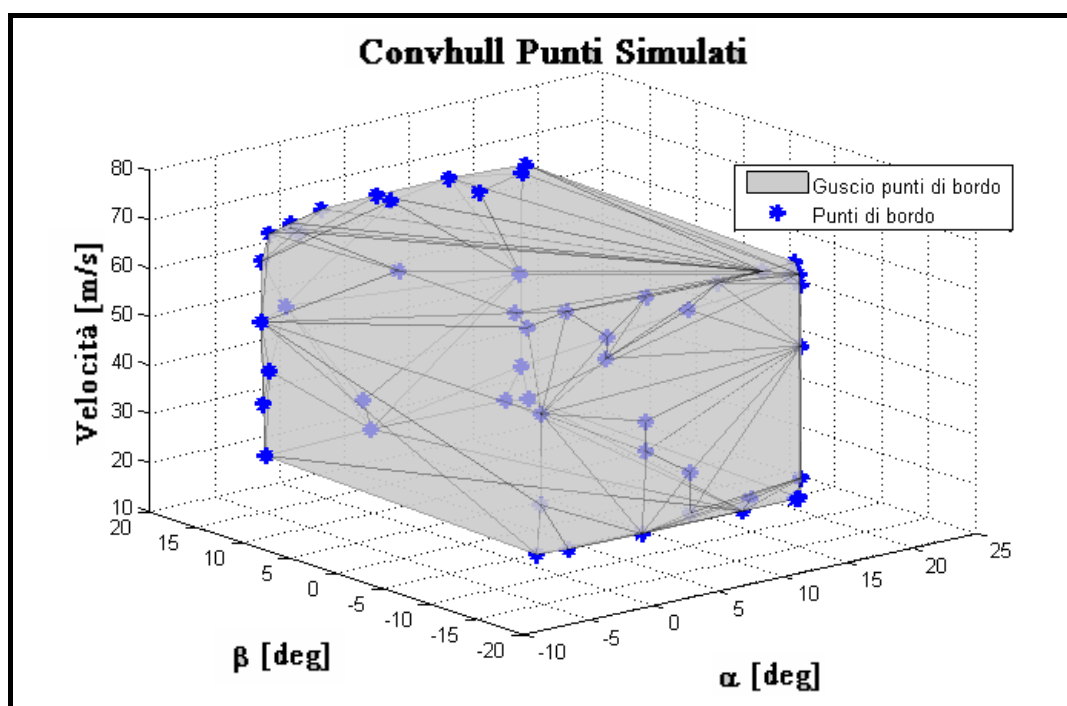


Fig. 5.5 Rappresentazione punti di bordo

Da lavori precedenti è stato valutato il rapporto tra il numero di punti per l'addestramento e il numero di punti per il *test*: in base al tale rapporto sono stati scelti, in maniera uniformemente distribuita nello spazio, 1010 punti per il *training set*, 8450 punti per *testing set* e 540 punti per *validation set* (vedi Fig. 5.6).

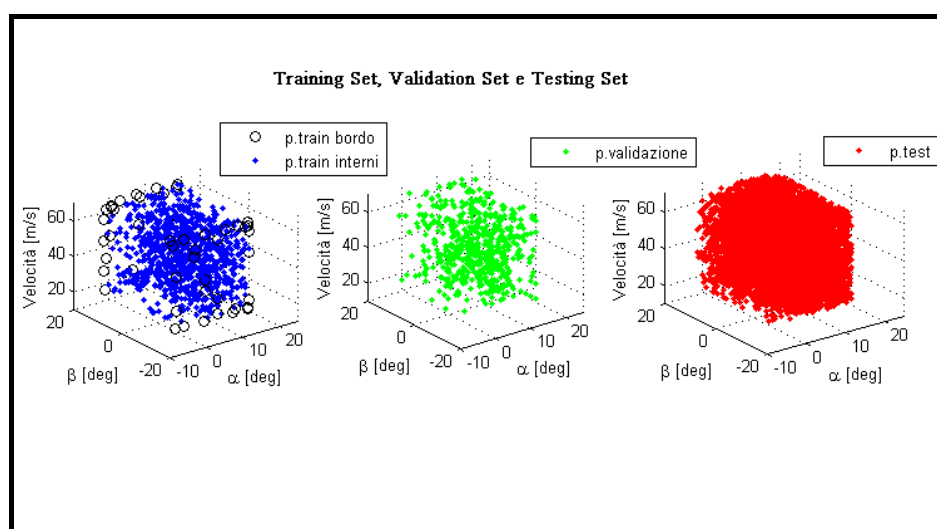


Fig. 5.6 Rappresentazione del Training set, Validation set e Testing set

Per fare in modo che i punti scelti risultino uniformemente distribuiti nello spazio, tutto il *set* di dati è stato diviso in *bin*, ovvero in contenitori cubici con dimensioni assegnate. In questo modo è possibile fare una stima della distribuzione della densità dei punti di simulazione all'interno di ogni *bin* e prelevare, con il rapporto assegnato in precedenza, i punti da destinare ai tre *data set*.

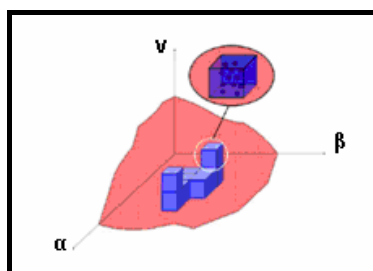


Fig. 5.7 Rappresentazione dei bin

Successivamente sono state prese in esame alcune possibili strutture neurali, variando il numero di *layer*, il numero di neuroni in ogni strato, le funzioni di trasferimento da attribuire ad ogni neurone, il tipo e il numero di ingressi da assegnare alla rete [6]. Per quanto riguarda quest'ultimo problema in primo luogo sono stati individuati i punti, tra gli 85 a disposizione sulla calotta sferica, che rendono la rete sensibile alla ricostruzione dei parametri di volo. Successivamente è stato fatto uno studio di ottimizzazione della posizione delle prese di misura, monitorando il valore della pressione in ognuno dei sensori al variare di α e di β , vedi *Fig. 5.9* e *Fig. 5.10* in cui sono rappresentati gli andamenti della pressione in alcuni punti. Gli andamenti in *Fig. 5.9* sono relativi a simulazioni effettuate ad $\alpha=0^\circ$, velocità=50 m/s, al variare di β , quelli in *Fig. 5.10* sono relativi a $\beta=0^\circ$, velocità=50 m/s, al variare di α . Da questa analisi è emerso che i punti più sensibili ai ΔP al variare degli angoli d'incidenza e derapata sono quelli visualizzati con un pallino nero numerato in *Fig. 5.8*: in questa figura sono rappresentati inoltre tutti i punti sui quali sono state monitorate le pressioni con dei pallini numerati di colore chiaro ed utilizzati come ingresso per le nostre reti [9]. A questi punti è stata assegnata una numerazione per poterli identificare.

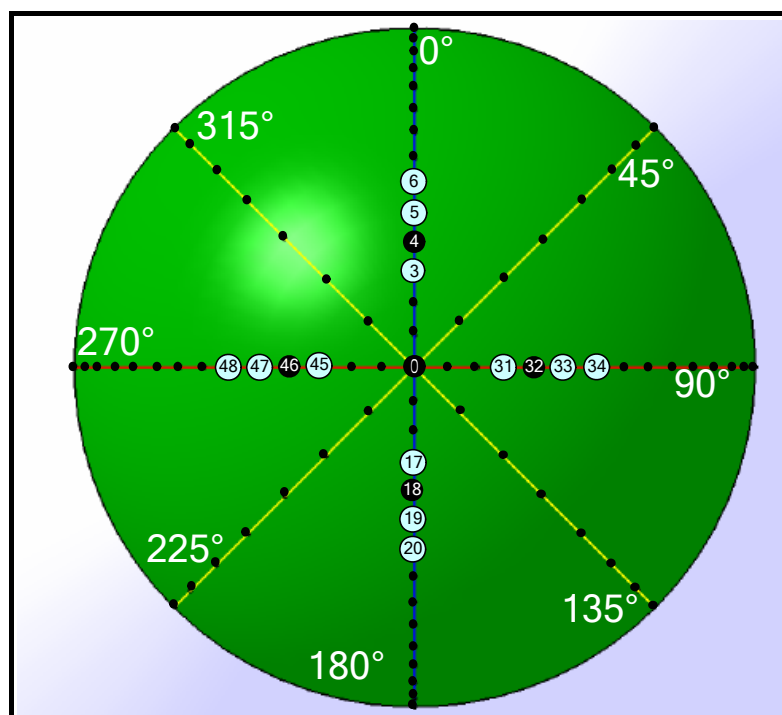


Fig. 5.8 Punti scelti come sensori per la sonda

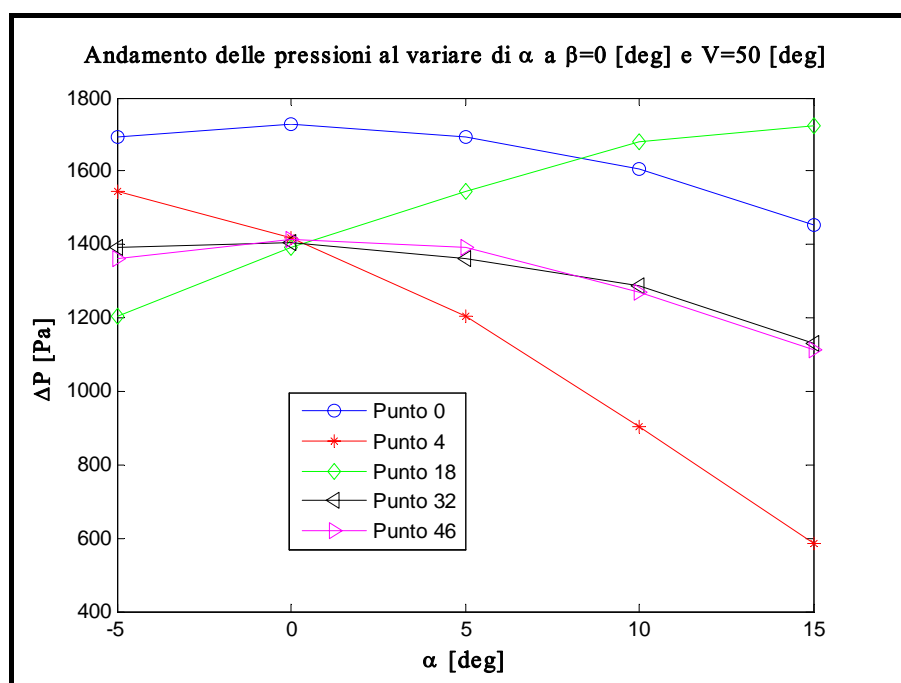


Fig. 5.9 Andamento delle pressioni al variare di α nei punti scelti per posizionare i sensori di misura

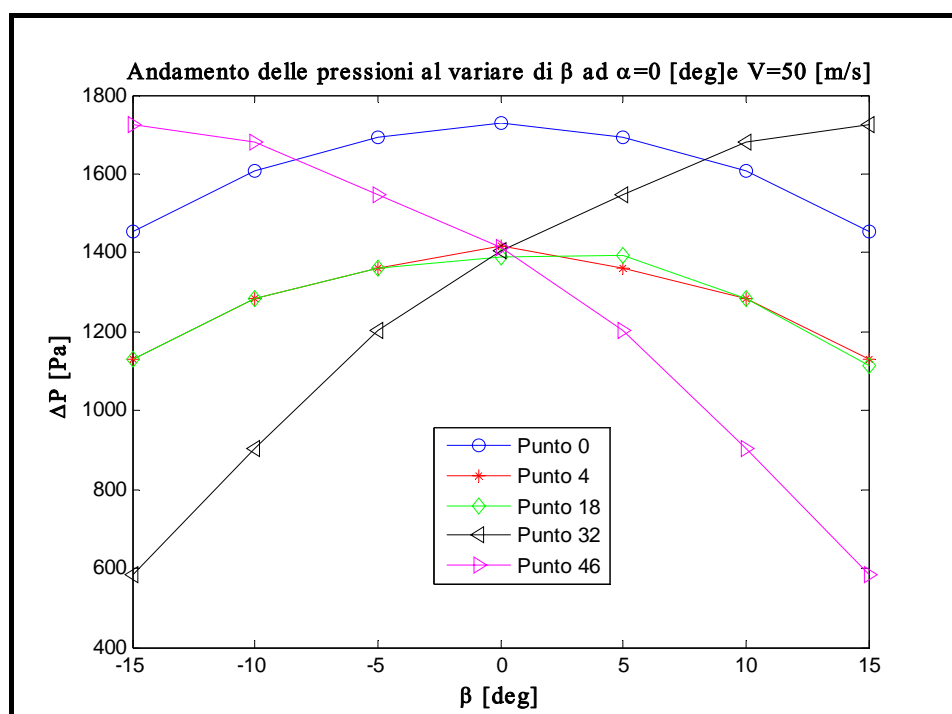


Fig. 5.10 Andamento delle pressioni al variare di β nei punti scelti per posizionare i sensori di misura

Scelti i punti i cui valori di pressione vengono utilizzati come *input* alle reti, si è passati a definire l'architettura delle reti. La scelta del numero di *layer* e del numero di neuroni per ogni *layer* è stata fatta in base alle esperienze maturate in precedenza, in particolare sono stati inseriti due strati nascosti con dieci neuroni ciascuno. Naturalmente le reti presenteranno anche un *layer* di uscita che avrà un numero di neuroni uguale al numero di *output* scelto per la rete. Per quanto riguarda la funzione di trasferimento da assegnare ai nodi la scelta è stata più delicata. Le principali funzioni di trasferimento prese in esame sono riportate di seguito [7]:

- **Funzione a gradino con soglia θ**

La funzione gradino con soglia θ converte l'ingresso in un segnale binario di tipo $[0,1]$, Fig. 5.11.

$$y = f(a) = \begin{cases} 0 \rightarrow a \geq \theta \\ 1 \rightarrow a < \theta \end{cases} \quad (\text{Eq. 5.1})$$

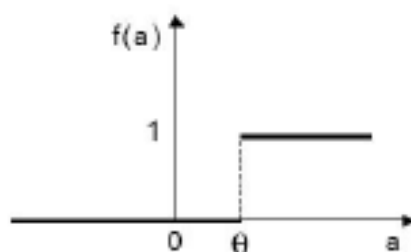


Fig. 5.11 Funzione heaviside

- **Funzione lineare**

La funzione lineare restituisce come uscita l'ingresso i come si vede in Fig. 5.12.

$$y = f(a) = a \quad (\text{Eq. 5.2})$$

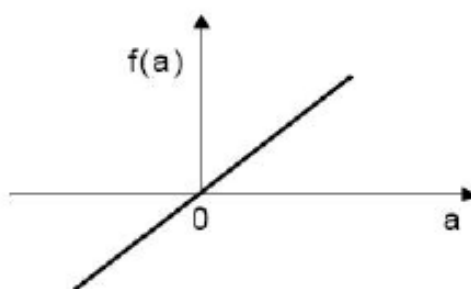


Fig. 5.12 Funzione lineare

- **Funzione lineare a tratti**

La funzione lineare a tratti viene usata come la funzione lineare solo a differenza della precedente è limitata da un intervallo, all'esterno del quale si comporta come una funzione gradino, *Fig. 5.13*.

$$y = f(a) = \begin{cases} 0 \rightarrow a < -0.5 \\ a + 0.5 \rightarrow -0.5 \leq a \leq +0.5 \\ 1 \rightarrow a > +0.5 \end{cases} \quad (\text{Eq. 5.3})$$

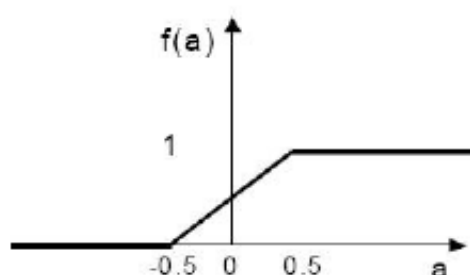


Fig. 5.13 Funzione lineare a tratti

- **Funzione sigmoide**

La funzione sigmoide è tra le più utilizzate. A differenza delle funzioni di soglia, che assumono solo i valori 0 e 1, la funzione sigmoide può assumere tutti i valori compresi in questo intervallo. In *Fig. 5.14* è rappresentata la funzione sigmoide logaritmica.

$$y = f(a) = \frac{1}{1 + e^{-T \cdot a}} \quad (\text{Eq. 5.4})$$

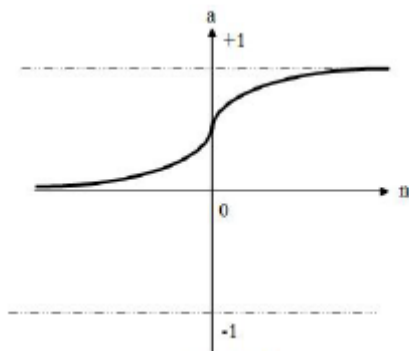


Fig. 5.14 Funzione sigmoide logaritmica

Si noti inoltre che queste funzioni sono governate da un parametro detto “Parametro di temperatura” che né controlla la pendenza.

- **Funzione iperbolica**

La funzione iperbolica produce valori d’uscita compresi tra -1 e 1 ed è anch’essa governata dal parametro di temperatura, vedi Fig. 5.15.

$$y = f(a) = \frac{e^{T \cdot a} - e^{-T \cdot a}}{e^{T \cdot a} + e^{-T \cdot a}} \quad (\text{Eq. 5.5})$$

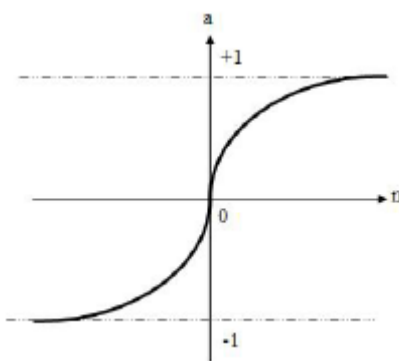


Fig. 5.15 Funzione iperbolica

6 Risultati e valutazione delle prestazioni delle reti

In questo capitolo sono descritte le prove eseguite in funzione delle strutture neurali scelte. Tutte le reti create sono di tipo *Multi Layer Perceptron* (MLP), con due strati nascosti con 10 neuroni ciascuno, e sono tutte state addestrate con l'algoritmo di tipo *Backpropagation - feed forward*. Inoltre, come da studi precedenti, la funzione di trasferimento utilizzata è la funzione sigmoide logaritmica con parametro di temperatura $T=4$ in maniera tale da aumentare la pendenza della funzione. Le principali tipologie di reti analizzate sono state quelle a 5 *input*, aventi come segnale d'ingresso il valore della pressione locale in 5 prese differenti. Naturalmente sono state provate solamente alcune tra tutte le possibili combinazioni di prese di pressione. La Tab. 6.1 riporta le tipologie di rete a 5 *input*.

Nome rete	N° Input	Input
<i>Net_SI_0_3_17_31_45</i>	5	$P_0, P_3, P_{17}, P_{31}, P_{45}$
<i>Net_SI_0_4_18_32_46</i>	5	$P_0, P_4, P_{18}, P_{32}, P_{46}$
<i>Net_SI_0_5_19_33_47</i>	5	$P_0, P_5, P_{19}, P_{33}, P_{47}$
<i>Net_SI_0_6_20_34_48</i>	5	$P_0, P_6, P_{20}, P_{34}, P_{48}$

Tab. 6.1 Prima tipologia di reti

La sigla *Simple Input*, SI, indica che gli *input* vengono inseriti nella rete così come sono presenti nel *database* aerodinamico, senza che venga effettuata alcuna normalizzazione per riportarli a valori compresi tra 0 e 1. Con P_j viene indicato il valore del ΔP misurato dalla j -esima presa. Inoltre si nota che la numerazione dei sensori va da 0 a 85 dove è indicato con 0 il sensore centrale nel punto d'incrocio dell'arco verticale con quello orizzontale. Con numeri da 1 a 28 sono indicati i punti sull'arco verticale, con numeri da 29 a 56 i punti sull'arco orizzontale e con

numeri da 57 a 85 tutti i punti sui due archi inclinati di 45° gradi. A partire da questo gruppo di 4 reti è stato successivamente creato un altro gruppo che, analogamente al primo, ha come *input* i valori dei ΔP misurati negli stessi punti delle reti precedenti, questa volta normalizzati attraverso il valore massimo e minimo di pressione riscontrato in tutto l'involuppo di volo. Il generico *input* normalizzato delle reti risulta essere dunque

$$i^N = \frac{i - \min(i(P_0))}{\max(i(P_0)) - \min(i(P_0))} \quad (Eq. 6.1)$$

dove $\max(i(P_0))$ e $\min(i(P_0))$ sono il valore massimo e minimo della pressione riferita al punto centrale 0. In questo modo si riconduce tutto l'intervallo di dati ad un intervallo compreso tra 0 e 1. Così facendo si è creato un nuovo gruppo di reti che sono state nominate come in *Tab. 6.2*.

Nome rete	N° Input	Input
<i>Net_NSI_0_3_17_31_45</i>	5	$P_0, P_3, P_{17}, P_{31}, P_{45}$
<i>Net_NSI_0_4_18_32_46</i>	5	$P_0, P_4, P_{18}, P_{32}, P_{46}$
<i>Net_NSI_0_5_19_33_47</i>	5	$P_0, P_5, P_{19}, P_{33}, P_{47}$
<i>Net_NSI_0_6_20_34_48</i>	5	$P_0, P_6, P_{20}, P_{34}, P_{48}$

Tab. 6.2 Seconda tipologia di reti

In tutto quindi sono state provate 8 reti con differenti *input* ma con la medesima architettura.

6.1 Parametri di confronto delle reti neurali

Le reti neurali citate in precedenza sono state provate classificandole secondo un ulteriore parametro che è il numero di *output*. Sono state utilizzate reti ad un solo neurone d'uscita impostando come *output* l'angolo d'incidenza α o l'angolo di derapata β oppure la velocità del velivolo in m/s. Di conseguenza sono stati creati 3 gruppi da 8 reti a singola uscita ciascuno. Ciascuno di questi 3 gruppi è stato addestrato sia con l'algoritmo di training *Bayesian Regularization* sia con l'algoritmo *Early Stopping*. In Fig. 6.1 è riportata e chiarita la notazione usata per le reti:

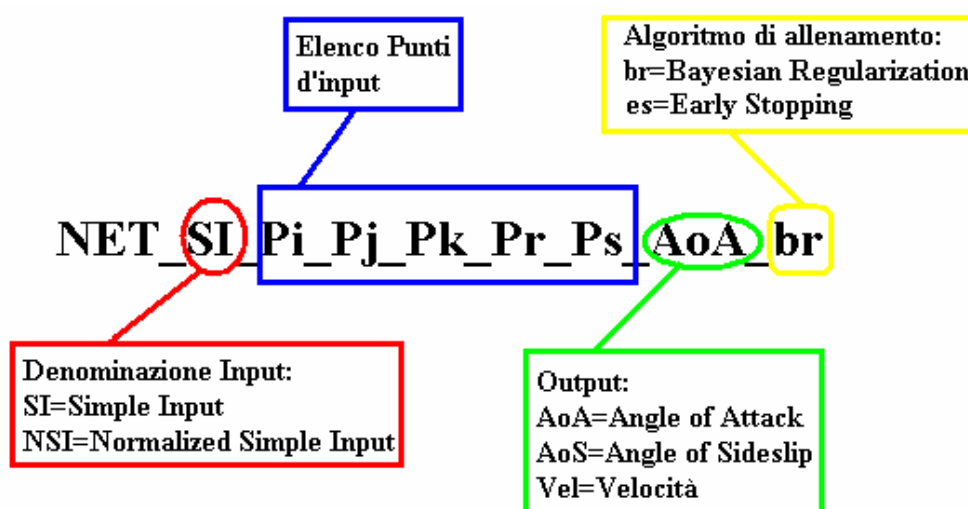


Fig. 6.1 Denominazione delle reti

Per quanto riguarda i risultati sono state prese in esame sia quelli relativi alla fase di addestramento che quelli relativi alla fase di *testing*. Nella fase di addestramento, come già detto in precedenza, il parametro che si considera principalmente come indice di buon funzionamento della rete è l'errore che questa commette durante il *training*. Per il metodo *Bayesian Regularization* l'errore viene calcolato come somma dei quadrati degli errori commessi in ogni punto di

addestramento (*Sum of Squared Weight*, SSE); invece il metodo *Early Stopping* utilizza il *Mean of Squared Wheight* (MSE), che viene calcolato come la media degli errori al quadrato commessi nei punti di addestramento. Sulla base delle esperienze maturate presso il DIA si è utilizzato un numero massimo di epoche pari a 3000, un valore del “goal” (ovvero del valore della cifra di merito sotto del quale si deve fermare l’addestramento) di 10^{-5} e un valore del parametro μ di direzione di 10^{-3} : tali impostazioni sono state mantenute per tutte le reti utilizzate. Inoltre subito dopo l’addestramento viene eseguita una simulazione utilizzando come *input* lo stesso *Training set*. La risposta della rete viene comparata con quello che dovrebbe essere l’uscita voluta, calcolando così l’errore. Poiché la funzione di trasferimento di ogni neurone fornisce una risposta compresa in un intervallo tra 0 e 1 l’*output* della rete viene fornito sotto forma di vettore a n righe, dove n è il numero di punti del *database di training*. A partire da tale vettore, confrontandolo con il valore delle uscite desiderate, viene calcolato un vettore degli errori. Dopo aver denormalizzato il vettore degli errori, si ottengono 2 parametri, vedi Fig. 6.2.

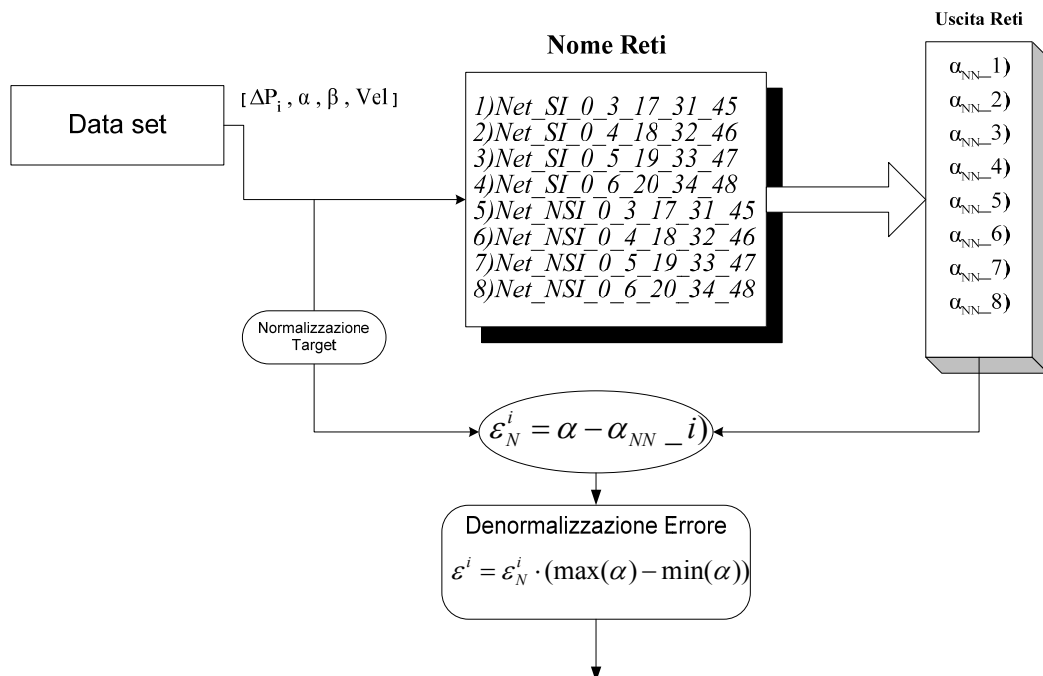


Fig. 6.2 Schema di calcolo dell’errore commesso dalle reti

Il primo parametro che si considera è il valore massimo dell'errore commesso sul *set* di *training* e il secondo parametro è il valore dell'errore medio, calcolato come somma degli errori in tutti i punti della simulazione diviso il numero di punti.

Dopo l'addestramento viene effettuata una simulazione sul *testing set* per valutare la capacità delle reti di generalizzare su punti al di fuori del *training set*. Anche da questo *test*, con la stessa procedura effettuata sul *set* di *training*, vengono calcolati l'errore medio e l'errore massimo. Inoltre si considerano altri 2 parametri: il numero di volte che l'errore commesso è maggiore di un certo valore e la distribuzione di questi picchi dell'errore nello spazio di α , β e velocità.

6.2 Reti a singola uscita

Come già detto in precedenza, queste tipologie di reti hanno l'obiettivo di individuare gli errori commessi nel calcolo dell'angolo d'attacco, dell'angolo di derapata o della velocità.

Il *database* di *training* è composto da 1010 punti di simulazione calcolati all'interno dell'involuppo di volo considerato. Il dominio degli *output* considerato è $[-5^\circ, 15^\circ]$ per l'angolo d'incidenza, $[-15^\circ, 15^\circ]$ per l'angolo di derapata e $[20 \text{ m/s}, 70 \text{ m/s}]$ per la velocità. Nelle reti addestrate con il metodo *Early stopping* viene utilizzato anche il *database* di validazione, composto da ulteriori 540 punti, con il quale, durante l'addestramento, viene fatta una simulazione e valutati gli errori. Per il *testing* invece viene utilizzato un *set* di dati di 8550 punti. Inoltre sulle reti che usano il metodo di addestramento *Bayesian Regularization* è stata fatta un'ulteriore prova utilizzando come dati di addestramento il *training set* ampliato con il *validation set*, ottenendo un *database* di 1550 punti totali. Tale prova è stata effettuata solo per le reti che hanno come *output* la velocità per valutare l'influenza del *database* di addestramento sul valore dei picchi dell'errore. Nelle *Tab. 6.3*, *Tab. 6.4* e *Tab. 6.5* sono stati inseriti tutti i risultati ottenuti durante le prove.

Output	Input	Metodo di allenamento	Training set		Testing set	
			SSE/MSE [deg]	Errore Massimo [deg]	Errore Medio [deg]	Errore Massimo [deg]
α	Net_SI_Pi_Pj_Pk_Pr_Ps_AoA_br					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,017306	0,447420	0,061806	4,708000
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,009211	0,309000	0,044788	4,314300
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,018390	0,652410	0,058578	11,619000
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,019264	0,557770	0,064088	2,997800
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoA_br					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,011873	0,427340	0,047425	2,701900
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,022973	0,653120	0,063271	11,798000
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,015694	0,581250	0,053015	3,570400
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,012754	0,437120	0,049785	3,178600
	Net_SI_Pi_Pj_Pk_Pr_Ps_AoA_es					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000941	3,749000	0,471450	17,255000
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,00050071	3,2339	0,31845	7,9663
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000399	2,721100	0,298050	6,499800
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,006979	6,958300	1,269900	7,660600
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoA_es					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000062	0,958130	0,110110	3,568700
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000202	1,932400	0,175050	4,524000
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,00019901	2,5759	0,18295	4,6833
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000377	3,190300	0,267930	4,749000

Tab. 6.3 Tabella degli errori commessi dalle reti a singolo output α

Output	Input	Metodo di allenamento	Training set		Testing set	
			SSE/MSE [deg]	Errore Massimo [deg]	Errore Medio [deg]	Errore Massimo [deg]
β	Net_SI_Pi_Pj_Pk_Pr_Ps_AoS_br					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,011152	0,764020	0,069483	4,455600
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,011031	0,594620	0,070484	9,266900
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,008324	0,598160	0,063145	4,057400
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,020640	0,760260	0,103840	23,727000
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoS_br					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,011685	0,776740	0,069462	5,347000
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,009937	0,574170	0,066164	7,574800
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,006967	0,609290	0,055964	10,033000
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,006619	0,379380	0,055509	2,526300
	Net_SI_Pi_Pj_Pk_Pr_Ps_AoS_es					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000875	5,198600	0,924730	11,976000
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000444	3,331600	0,470510	15,160000
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000116	3,178600	0,212100	5,204100
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000743	3,096400	0,608880	4,601700
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoS_es					
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000210	3,825400	0,293820	25,922000
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000300	3,817700	0,385740	5,116200
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000165	3,243200	0,259680	5,259700
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000122	3,054500	0,218050	5,817700
						0,373280
						0,417430
						0,301640
						0,237200

Tab. 6.4 Tabella degli errori commessi dalle reti a singolo output β

Output	Input	Metodo di allenamento	Training set			Testing set		
			SSE/MSE [m/s]	Errore Massimo [m/s]	Errore Medio [m/s]	Errore Massimo [m/s]	Errore Medio [m/s]	
Vel	Net_SI_Pi_Pj_Pk_Pr_Ps_Vel_br							
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,008700	0,874600	0,110300	12,391000	0,140410	
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	85,050300	24,710000	12,516000	24,781000	12,277000	
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,045200	1,057700	0,242900	29,735600	0,508900	
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	86,266000	25,086600	12,600200	25,080000	12,353000	
	Net_NSI_Pi_Pj_Pk_Pr_Ps_Vel_br							
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,008693	0,141600	0,035600	0,854880	0,050528	
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,008850	0,158100	0,037000	0,547790	0,045950	
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,009110	0,159400	0,037700	0,497110	0,049604	
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,007840	0,133400	0,034400	0,725550	0,046050	
	Net_SI_Pi_Pj_Pk_Pr_Ps_Vel_es							
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000068	2,166500	0,309160	4,631300	0,355100	
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000010	0,883550	0,115870	4,066600	0,177210	
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000001	0,92695	0,13192	1,7924	0,15378	
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000249	3,539800	0,640040	9,267400	0,754450	
	Net_NSI_Pi_Pj_Pk_Pr_Ps_Vel_es							
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000031	1,538100	0,208610	2,493600	0,240780	
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000010	0,734260	0,122780	4,325600	0,168680	
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000010	1,151000	0,116330	2,091700	0,157390	
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000024	1,742800	0,216580	3,134000	0,267240	

Tab. 6.5 Tabella degli errori commessi dalle reti a singolo output Velocità

Le tabelle sono divise in due gruppi, nel primo sono stati inseriti i risultati relativi all'addestramento effettuato tramite il metodo *Bayesian Regularization*, invece il secondo gruppo riporta i risultati delle reti addestrate con il metodo *Early Stopping*. Inoltre un'ulteriore divisione è stata fatta differenziando gli errori commessi durante la fase di *training* e quelli commessi durante il *testing*. Sono riportati sia gli errori medi che gli errori massimi commessi sia durante l'addestramento che durante il *testing*. Dai risultati relativi alle reti che hanno come uscita l'angolo d'incidenza si evince che:

- Durante l'addestramento gli errori commessi si abbassano notevolmente arrivando praticamente per tutte le reti a ordini di grandezza di 10^{-2} nel caso del metodo *Bayesian Regularization* e fino ad arrivare a 10^{-4} nel caso delle reti addestrate col metodo *Early Stopping*, ES, ricordando che in questo caso il parametro di errore considerato è il MSE.
- Le reti addestrate con il metodo ES gli errori medi e massimi sul *training set* sono dell'ordine rispettivamente di 10^{-1} e di $1 \div 3$ gradi; mentre sul *testing set* gli errori medi sono dell'ordine di $0.5 \div 2$ gradi e quelli massimi di $3 \div 6$ con picchi per alcune reti anche di $18 \div 20$ gradi.
- Nel caso delle reti addestrate col metodo BR in fase di addestramento gli errori medi e massimi sono dell'ordine rispettivamente di $4 \cdot 10^{-2}$ e $4 \cdot 10^{-1}$. Durante il *test* gli errori massimi e medi sono più bassi che nel caso delle reti addestrate con il metodo ES ed hanno ordini di grandezza di $6 \div 9 \cdot 10^{-2}$ per l'errore medio e $2 \div 4$ gradi per l'errore massimo, con picchi di $10 \div 11$ gradi per qualche rete.

Non ci sono differenze sostanziali tra le reti per quanto riguarda la scelta dei punti di *input* evidenziando che la scelta della zona dove sono posizionate le prese di pressione è stata fatta adeguatamente e che quindi sarebbe possibile realizzare delle ridondanze facendo funzionare in simultanea reti con diverse prese di pressione posizionate molto vicino.

Inoltre visualizzando la distribuzione dell'errore in funzione di α , β e velocità si vede che i picchi dell'errore sono spesso a bassa velocità ed in generale sono distribuiti sui bordi dell'involuppo di volo (vedi Fig. 6.3).

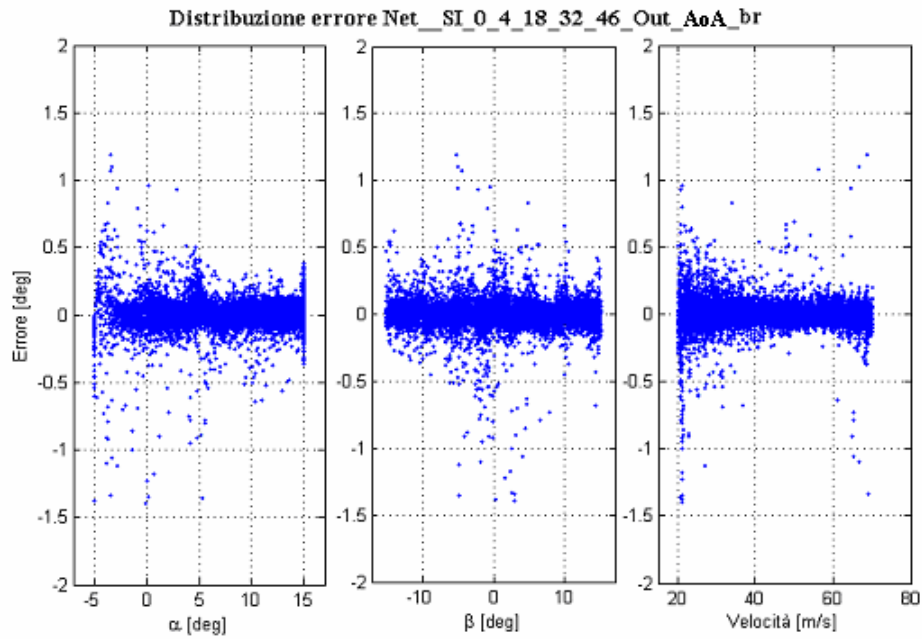


Fig. 6.3 Distribuzione degli errori nella stima di α in funzione di α , β e velocità

A titolo di esempio la Fig. 6.3 riporta gli errori relativi alla rete “Net_SI_P0_P4_P18_P32_P46_AoA_br”, che è risultata la migliore tra tutte quelle esaminate in base all'errore medio calcolato sul *testing set*.

Un'altra sostanziale differenza tra le reti addestrate col metodo ES e le reti addestrate con il BR è il numero di eccedenze dell'errore, ovvero il numero di volte che l'errore è maggiore di 0.5: per le prime il numero delle eccedenze è dell'ordine di qualche migliaia e per le seconde si aggira intorno a qualche centinaia, vedi Tab. 6.6. Da questo dato si spiega la presenza di un errore medio decisamente maggiore per le reti addestrate con il metodo ES.

Nome Rete	Metodo di addestramento	Eccedenze	Errore Max Percentuale (%)
1)Net_NSI_0_3_17_31_45_Out_AoA_br	Bayesian Regularization	83	13,51
2)Net_NSI_0_4_18_32_46_Out_AoA_br		146	58,99
3)Net_NSI_0_5_19_33_47_Out_AoA_br		148	17,85
4)Net_NSI_0_6_20_34_48_Out_AoA_br		275	15,89
5)Net_SI_0_3_17_31_45_Out_AoA_br		157	23,54
6)Net_SI_0_4_18_32_46_Out_AoA_br		80	21,57
7)Net_SI_0_5_19_33_47_Out_AoA_br		134	58,10
8)Net_SI_0_6_20_34_48_Out_AoA_br		100	14,99
1)Net_NSI_0_3_17_31_45_Out_AoA_es	Early Stopping	434	17,84
2)Net_NSI_0_4_18_32_46_Out_AoA_es		765	22,62
3)Net_NSI_0_5_19_33_47_Out_AoA_es		2603	23,42
4)Net_NSI_0_6_20_34_48_Out_AoA_es		1682	23,75
5)Net_SI_0_3_17_31_45_Out_AoA_es		3925	86,28
6)Net_SI_0_4_18_32_46_Out_AoA_es		7672	39,83
7)Net_SI_0_5_19_33_47_Out_AoA_es		2670	32,50
8)Net_SI_0_6_20_34_48_Out_AoA_es		6697	38,30

Tab. 6.6 Eccedenze ed errore percentuale relativo alle reti a singola uscita in α

Per quanto riguarda l'errore commesso nel calcolo dell'angolo di derapata β si possono fare le seguenti considerazioni:

- Parametri di addestramento SSE più bassi rispetto a quelli delle reti con uscita in α arrivando anche a $0.6 \cdot 10^{-3}$.
- Per le reti addestrate con il metodo BR, errori medi sul *training* dell'ordine di 10^{-2} gradi ed errori massimi di circa 0.5 gradi. Sul *testing set*, errori medi di circa 10^{-1} gradi ed errori massimi di 4÷5 gradi con alcuni picchi anche di 10 gradi.
- Le reti addestrate con il metodo ES invece presentano, come nel caso delle reti con uscita α , MSE molto bassi dell'ordine di 10^{-4} , errori medi e massimi calcolati sul *training set* di rispettivamente 0.3÷0.5 gradi e 3÷5 gradi ed errori medi e massimi calcolati sul *testing set* di rispettivamente

di $0.3 \div 1$ gradi e $3 \div 5$ gradi con picchi isolati di $11 \div 15$ gradi per l'errore massimo.

Per quanto riguarda il gruppo di reti utilizzate per il calcolo della velocità di volo, i risultati sono sostanzialmente simili agli altri due gruppi. Si nota che a volte i risultati del *test* portino a degli errori massimi particolarmente elevati, anche di $20 \div 25$ gradi, ma bisogna considerare che questi picchi dell'errore sono dovuti spesso a problemi riscontrati durante l'addestramento (ad esempio un'inizializzazione dei pesi sbagliata). Questi problemi però sono anche facilmente risolvibili semplicemente rieseguendo l'addestramento o infittendo il *training set* con dei punti di bordo e addestrando nuovamente le reti.

Ad esempio alcune reti, addestrate con il metodo BR, per la ricostruzione della velocità, commettono errori elevati sia durante l'addestramento che durante il *test*. Riaddestrando queste reti, con un *set* di *training* ampliato mediante il *validation set*, si ottengono degli errori medi e massimi, sia sul *training set* che sul *testing set*, notevolmente più bassi. In questo modo si eliminano i picchi dell'errori riscontrati in precedenza, vedi Tab. 6.7. Questa tipologia di problemi relativi all'addestramento, si possono facilmente individuare monitorando il parametro d'addestramento SSE/MSE.

Output	Input	Metodo di allenamento	Training set			Testing set	
			SSE/MSE [m/s]	Errore Massimo [m/s]	Errore Medio [m/s]	Errore Massimo [m/s]	Errore Medio [m/s]
Vel	Net_SI_Pi_Pj_Pk_Pr_Ps_Vel_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,0015772	0,15862	0,03993	1,861000	0,048795
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,001515	0,174390	0,039138	1,905900	0,044708
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,002999	0,244490	0,055122	0,478380	0,059119
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,0014504	0,20947	0,037899	0,403620	0,043336
	Net_NSI_Pi_Pj_Pk_Pr_Ps_Vel_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,001492	0,168170	0,038934	0,598960	0,043529
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,001412	0,188160	0,037157	0,503690	0,043686
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,001601	0,201770	0,040142	0,801010	0,045888
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,001099	0,156190	0,032667	0,665400	0,037550

Tab. 6.7 Errori reti ad singola uscita (Vel) riaddestrate con training set ampliato

I picchi dell'errore sono di solito localizzati nei punti di bordo dell'involuppo di volo e specialmente come si vede in *Fig. 6.4* e *Fig. 6.5* a bassa velocità, analogamente a quanto accadeva nelle reti con uscita α . In *Tab. 6.8* e *Tab. 6.9* si riporta come per le reti analizzate in precedenza il numero delle eccedenze e l'errore percentuale massimo. Di seguito è riportata la distribuzione degli errori della rete che ha raggiunto i risultati migliori per quanto riguarda l'errore medio sul *testing set*. Le eccedenze sono conteggiate come il numero di volte che l'errore supera 0,5 gradi per l'angolo di derapata β e 0,1 m/s per la velocità di volo.

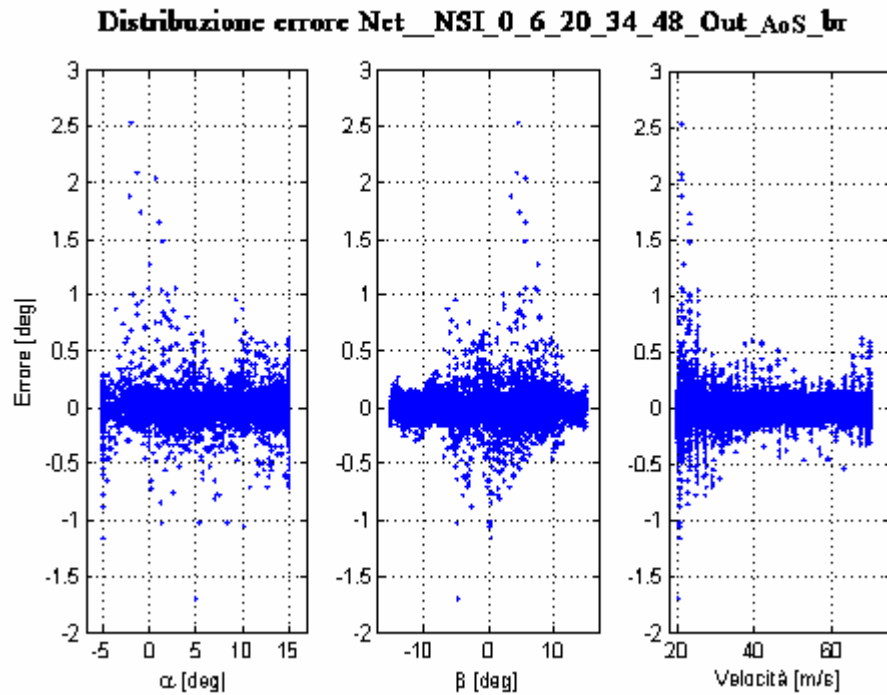
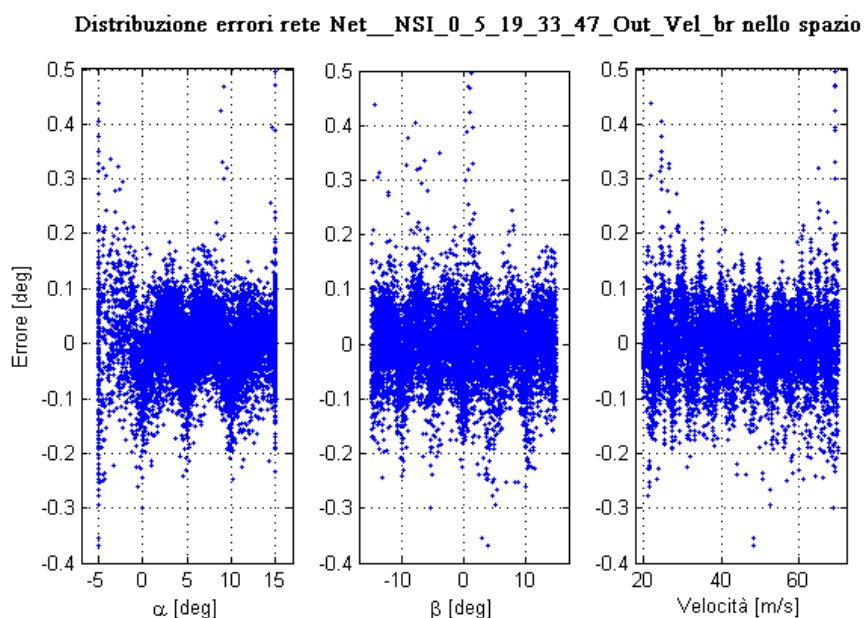


Fig. 6.4 Distribuzione degli errori nella stima di β in funzione di α , β e velocità

Nome Rete	Metodo di addestramento	Eccedenze	Errore Max Percentuale (%)
1)Net_NSI_0_3_17_31_45_Out_AoA_br	Bayesian Regularization	83	13,51
2)Net_NSI_0_4_18_32_46_Out_AoA_br		146	58,99
3)Net_NSI_0_5_19_33_47_Out_AoA_br		148	17,85
4)Net_NSI_0_6_20_34_48_Out_AoA_br		275	15,89
5)Net_SI_0_3_17_31_45_Out_AoA_br		157	23,54
6)Net_SI_0_4_18_32_46_Out_AoA_br		80	21,57
7)Net_SI_0_5_19_33_47_Out_AoA_br		134	58,10
8)Net_SI_0_6_20_34_48_Out_AoA_br		100	14,99
1)Net_NSI_0_3_17_31_45_Out_AoA_es	Early Stopping	1860	86,41
2)Net_NSI_0_4_18_32_46_Out_AoA_es		2470	17,05
3)Net_NSI_0_5_19_33_47_Out_AoA_es		1404	17,53
4)Net_NSI_0_6_20_34_48_Out_AoA_es		965	19,39
5)Net_SI_0_3_17_31_45_Out_AoA_es		5794	39,92
6)Net_SI_0_4_18_32_46_Out_AoA_es		3690	50,53
7)Net_SI_0_5_19_33_47_Out_AoA_es		881	17,35
8)Net SI 0 6 20 34 48 Out AoA es		4091	15,34

Tab. 6.8 Eccedenze ed errore percentuale delle reti a singola uscita in β Fig. 6.5 Distribuzione degli errori nella stima della velocità in funzione di α , β e velocità

Nome Rete	Metodo di addestramento	Eccedenze	Errore Max Percentuale (%)
1)Net_NSI_0_3_17_31_45_Out_Vel_br	Bayesian Regularization	987	1,71
2)Net_NSI_0_4_18_32_46_Out_Vel_br		752	1,10
3)Net_NSI_0_5_19_33_47_Out_Vel_br		903	0,99
4)Net_NSI_0_6_20_34_48_Out_Vel_br		718	1,45
5)Net_SI_0_3_17_31_45_Out_Vel_br		4190	24,78
6)Net_SI_0_4_18_32_46_Out_Vel_br		8451	49,56
7)Net_SI_0_5_19_33_47_Out_Vel_br		6306	59,47
8)Net_SI_0_6_20_34_48_Out_Vel_br		8422	50,16
1)Net_NSI_0_3_17_31_45_Out_Vel_es	Early Stopping	5836	4,99
2)Net_NSI_0_4_18_32_46_Out_Vel_es		4623	8,65
3)Net_NSI_0_5_19_33_47_Out_Vel_es		4563	4,18
4)Net_NSI_0_6_20_34_48_Out_Vel_es		6252	6,27
5)Net_SI_0_3_17_31_45_Out_Vel_es		6675	9,26
6)Net_SI_0_4_18_32_46_Out_Vel_es		4803	8,13
7)Net_SI_0_5_19_33_47_Out_Vel_es		4582	3,58
8)Net SI 0 6 20 34 48 Out Vel es		7602	18,53

Tab. 6.9 Eccedenze ed errore percentuale delle reti a singola uscita nella velocità

Le reti che ricostruiscono la velocità di volo hanno un errore percentuale molto più basso di quello delle altre reti. Questo deriva dal fatto che l'errore percentuale è calcolato in base alla grandezza del dominio della variabile presa in esame: nel caso della velocità questo dominio è più grande.

6.3 Reti a doppia uscita

Nel presente paragrafo vengono esaminate le reti a doppio *output* che presentano 2 neuroni nello strato d'uscita. Questo tipo di reti sono state utilizzate per il calcolo contemporaneo dell'angolo d'attacco e dell'angolo di derapata. I due gruppi di 4 reti, già provate in precedenza ad una sola uscita, sono state modificate nell'architettura per aggiungere un neurone di *output*. Anche per queste reti i parametri presi in esame nel valutare la funzionalità della rete sono stati l'SSE, nel caso delle reti addestrate con il metodo BR, e l'MSE nel caso di quelle addestrate con il metodo ES. Sono state fatte inoltre due simulazioni sul *training set* e sul *testing set* per calcolare l'errore massimo e medio. Anche per queste reti la simulazione sul *testing set* è servita per valutare la capacità di generalizzare su punti al di fuori del *database* di addestramento; a differenza delle reti a singola uscita è stato necessario calcolare gli errori commessi sia nel calcolo di α che in quello di β . La Tab. 6.10 riporta tutti i parametri d'interesse:

Input	Metodo di training	Output	Training set			Output	Testing set			Training set			Testing set		
			SSE/MSE	Errore Massimo	Errore Medio		Errore Massimo	Errore Medio	Errore Massimo	Errore Medio	Errore Massimo	Errore Medio			
Net_SI_Pi_Pj_Pk_Pr_Ps_AoA_AoS_br															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	br		0,0945	0,8825	0,0902		3,5604	0,1229		1,6179	0,1461		4,2448	0,1607	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	α	0,1246	0,6762	0,1054		3,5968	0,1233	β	1,4743	0,1844		4,8618	0,2140	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	br		0,1342	0,9298	0,1188		4,7172	0,1627		1,7574	0,1636		4,4720	0,1927	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	br		0,0990	1,1211	0,0987		3,474	0,13		1,2961	0,1427		4,8859	0,1743	
Net_NSI_Pi_Pj_Pk_Pr_Ps_AoA_AoS_br															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	br		0,0834	0,8729	0,0898		2,6855	0,1050		1,4586	0,1293		5,2067	0,1585	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	α	0,1209	1,0149	0,1072		3,4694	0,1272	β	1,9711	0,1562		4,1049	0,1863	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	br		0,1168	1,1133	0,1033		4,0192	0,1254		1,5897	0,1465		3,0730	0,1582	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	br		0,0836	0,7603	0,0811		3,0312	0,1048		2,1881	0,1306		4,8454	0,1421	
Net_SI_Pi_Pj_Pk_Pr_Ps_AoA_AoS_es															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm		0,0007	4,3831	0,3757		9,8864	0,4178		10,4890	0,4219		13,9550	0,4921	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	α	0,0002	3,5472	0,2021		4,5654	0,2346	β	2,7069	0,2455		6,8439	0,2867	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm		0,0003	3,7392	0,2236		12,4690	0,2662		3,3663	0,3181		9,2975	0,4286	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm		0,0001	1,5817	0,1343		4,0469	0,1736		2,8616	0,2066		8,4399	0,2713	
Net_NSI_Pi_Pj_Pk_Pr_Ps_AoA_AoS_es															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm		0,0003	3,6668	0,2013		4,1985	0,2090		3,5316	0,2767		4,7644	0,2937	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	α	0,0002	3,0267	0,2000		4,4808	0,2278	β	3,0795	0,2337		5,1495	0,2569	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm		0,0004	3,7781	0,2819		4,6664	0,3165		3,3266	0,3302		5,6393	0,3556	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm		0,0003	3,2081	0,2095		5,2580	0,2599		3,4358	0,3619		18,8960	0,4724	

Tab. 6.10 Tabella errori commessi dalle reti a doppia uscita

Dalla *Tab. 6.10* si evincono i seguenti risultati:

- Per le reti addestrate con il metodo BR l'SSE è dell'ordine di 10^{-1} , mentre l'MSE relativo alle reti addestrate con il metodo ES, dell'ordine di 10^{-4} .
- Errore medio (calcolato sul *training set* per le reti addestrate con il BR) dell'ordine di 10^{-1} gradi per entrambe le uscite. Errori massimi compresi tra 0.6 e 1.1 per α e tra 1 e 2 gradi per β .
- Per le reti addestrate con l'ES, si hanno, sul *training set*, errori medi compresi tra 0.2 e 0.3 gradi ed errori massimi tra 3 e 4 gradi con picchi anche di 12. Non si evincono differenze tra le due uscite per quanto riguarda gli errori medi e massimi.
- Sul *testing set* le reti addestrate con il metodo BR commettono errori medi dell'ordine di 10^{-1} , per entrambe le uscite, ed errori massimi compresi tra 2.7 e 4.7 gradi per α , tra 3 e 5.2, per β .
- Anche per le reti addestrate con il metodo ES gli errori medi sul *testing set* sono uguali per le due uscite, ed in particolare sono compresi tra 0.2 e 0.4 gradi. Gli errori massimi, per α sono di circa 4 gradi con picchi isolati di 12, per β di circa 5 gradi, con picchi di 13÷18.

Inoltre, si sono monitorate, come nel caso delle reti a singola uscita, l'andamento degli errori in funzione di α di β e della velocità sia per gli errori commessi nella stima di α che quelli in β .

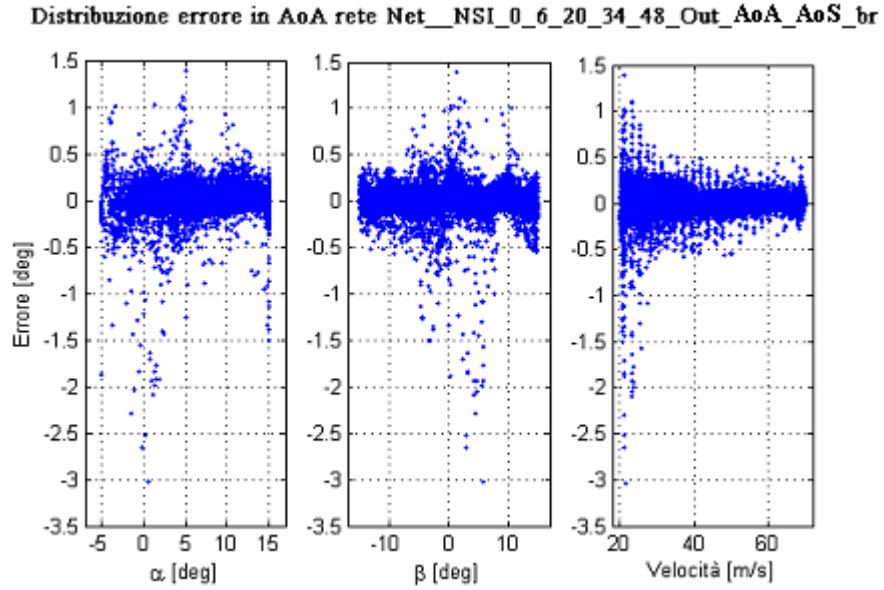


Fig. 6.6 Distribuzione degli errori nella stima di α in funzione di α , β e velocità per le reti a doppia uscita

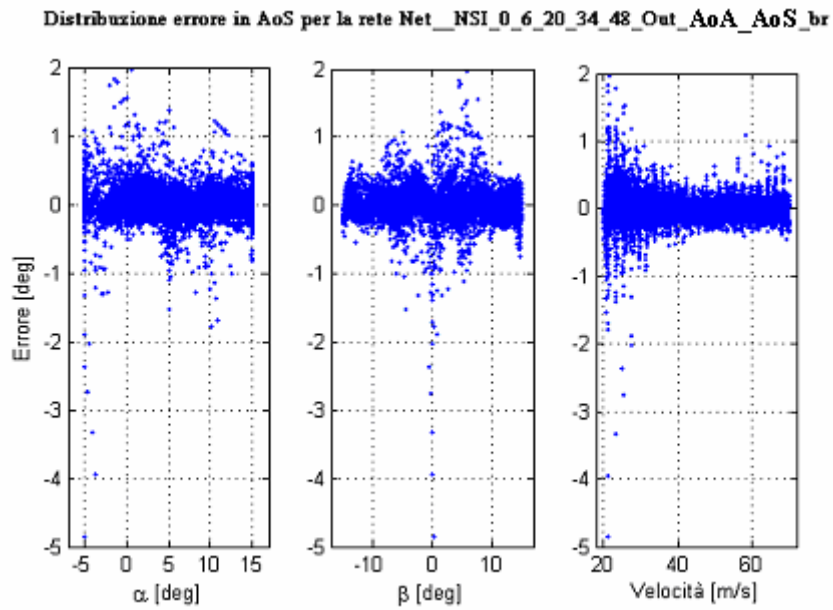


Fig. 6.7 Distribuzione degli errori nella stima di β in funzione di α , β e velocità per le reti a doppia uscita

Anche in questo tipo di rete si vede che gli errori più elevati sono posizionati sui punti di bordo dell'involuppo di volo, specialmente a bassa velocità.

6.4 Risultati generali

Dai risultati di tutte queste prove si è verificato che:

- La zona scelta per posizionare le prese di pressione ha prodotto buoni risultati fornendo un'adeguata sensibilità alle variazioni di α , β e velocità.
- La differenza di *input* tra le reti provate non ha cambiato sostanzialmente i risultati, generando per tutte e 4 le reti risultati ottimi.
- La normalizzazione degli *input* non ha cambiato l'ordine di grandezza degli errori.
- L'algoritmo di addestramento *Early Stopping* è molto più veloce rispetto al *Bayesian Regularization* poiché richiede un numero di epoche necessarie all'addestramento inferiore e arrivano ad un valore dell'MSE molto più basso del valore del SSE calcolato con il metodo BR.
- Il metodo BR funziona meglio rispetto al metodo ES, in quanto le reti addestrate con tale metodo riescono a generalizzare meglio su punti diversi dall'addestramento rispetto alle reti addestrate con il metodo ES.
- Le reti a singola uscita commettono errori medi e massimi sul *training set* più bassi rispetto a quelle a doppia uscita, nonostante queste ultime sul *testing set* forniscono errori medi e massimi più bassi.

Su quest'ultimo punto c'è da fare una precisazione. Infatti il numero di parametri di una rete neurale è uguale al numero di pesi più il numero di soglie d'attivazione dei neuroni:

$$N_{par} = N_{pesi} + N_{bias} \quad (Eq. 6.2)$$

Il numero di pesi dipende dal numero di neuroni in uscita ed il numero di *bias*, invece, è uguale al numero di neuroni. Per le reti a singola uscita il numero di parametri risulta essere di 181 e per le reti a doppia uscita di 192. Quindi il numero

di parametri per realizzare due reti a singola uscita è quasi il doppio rispetto al numero di parametri per realizzare una rete a doppia uscita. Questo porta ad un notevole risparmio in termini di memoria necessaria al sistema nell'utilizzo di reti a doppia uscita. Utilizzando reti a doppia uscita, però, è più difficile realizzare sistemi resistenti ad avarie in quanto perdendo un *input* si perderebbe la risposta su entrambe le uscite [3]. Riguardo alla fase di addestramento si vede bene dalle Fig. 6.8 e Fig. 6.9 come il metodo *Early Stopping* arrivi a valori del MSE molto bassi arrivando anche al “goal” in alcuni casi.

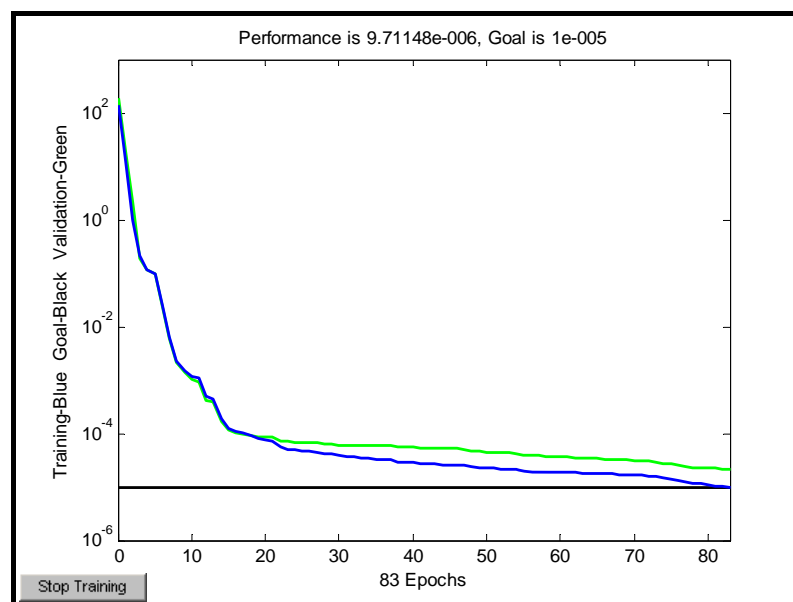


Fig. 6.8 Andamento MSE durante addestramento rete

Net_NSI_0_5_19_33_47_Out_Vel_es

Nella Fig. 6.8 è rappresentata la schermata di addestramento della rete “Net_NSI_0_5_19_33_47_Out_Vel_es” in cui si vede l’andamento del parametro MSE calcolato sul *training set*, in verde, e in blu lo stesso calcolato sul *validation set*. In alto inoltre è possibile notare il valore a cui è arrivato tale parametro nel momento in cui l’addestramento è stato fermato per il raggiungimento del “goal” dopo 83 epoche, in basso.

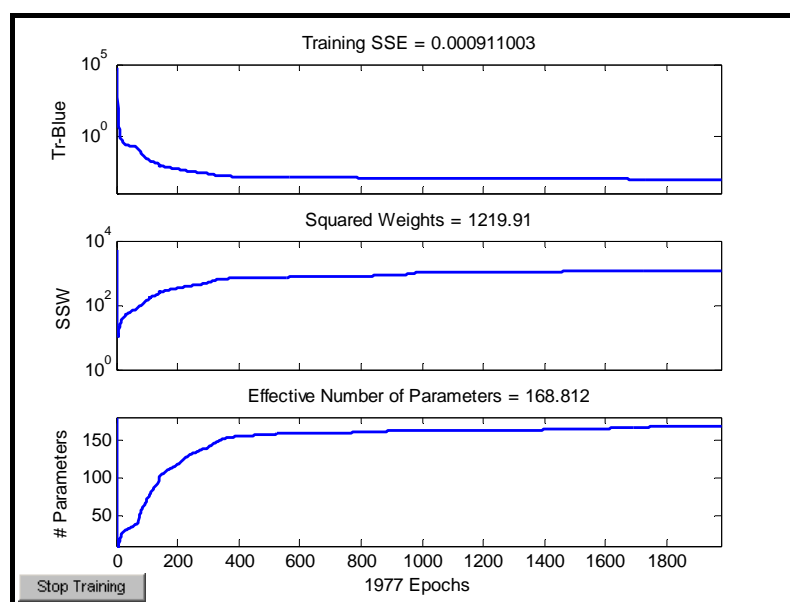


Fig. 6.9 Andamento SSE durante addestramento rete

Net_NSI_0_5_19_33_47_Out_Vel_br

Nel metodo *Bayesian Regularization* il parametro SSE non risulta mai più basso di 10^{-3} nonostante poi le reti addestrate con tale metodo generalizzano meglio rispetto alle altre. In Fig. 6.9 è visualizzata la schermata di addestramento della stessa rete di prima però addestrata con il metodo BR. In tale figura è possibile vedere oltre all'andamento del SSE, prima figura in alto, il valore minimo raggiunto dopo che l'addestramento è stato fermato a 1977 epoche, in basso al centro. Inoltre sono visualizzati gli andamenti del numero di parametri di cui la rete ha bisogno ed il valore di *Sum of Squared Weights* (SSW). Il numero di parametri di cui la rete a bisogno è inferiore al numero massimo calcolato nell'Eq. 6.2, perché durante l'addestramento la rete cerca di alleggerire la matrice dei pesi eliminando tutti quei valori che valuta non essere necessari.

7 Conclusioni e sviluppi futuri

Il lavoro di tesi ha portato alla definizione di una sonda multi-pressione che usa algoritmi di calcolo di tipo non convenzionale, basati su reti neurali. Lo studio complessivamente svolto è suddiviso in due fasi: la definizione di un database di pressioni preliminare attraverso l'analisi CFD e la caratterizzazione di algoritmi di calcolo basati su reti neurali. Nella prima fase è stata definita la geometria della sonda e sono state eseguite le simulazioni fluidodinamiche in tutto l'involuppo di volo preso in considerazione. Nella seconda fase sono state studiate le possibili architetture delle reti neurali e sono stati calibrati questi algoritmi di calcolo cercando di ottimizzarne le prestazioni.

Gli algoritmi di calcolo basati su reti neurali hanno prodotto ottimi risultati mettendo in evidenza la bontà dell'architettura scelta per la ricostruzione dei parametri α , β e velocità. Inoltre dalle simulazioni eseguite sulle varie tipologie di reti si sono riscontrati errori medi nel calcolo dei parametri di volo di qualche decimo di grado, con dei picchi di qualche grado solo in punti isolati.

In futuro verrà sviluppata una rete neurale capace di fornire una stima della pressione statica sulla base delle misure di pressione. Verranno effettuate delle prove in galleria del vento su un modello della sonda al fine di confrontare il *database* ottenuto a quello proveniente dall'analisi CFD. Attraverso queste prove in galleria si potrà, inoltre, svolgere una ricalibrazione degli algoritmi di ricostruzione dei parametri di volo.

Altre attività riguardano lo studio e lo sviluppo di configurazioni della sonda con un numero elevato di fori per la misura delle pressioni (10÷15). Tale configurazione a più fori permetterebbe di aumentare il livello di affidabilità della sonda stessa e su opportuni algoritmi di rilevamento delle avarie dei trasduttori di pressione si potranno svolgere delle ulteriori attività di studio.

Bibliografia

1. Fluent inc., “Fluent 6.1 User’s Guide, February 2003
2. Timothy R. Moes, Stephen A. Whitmore, “A preliminary look of techniques used to obtain Airdata from flight at high angle of attack, NASA Technical Memorandum, December 1990.
3. C.V. Srinatha Sastry, K.S. Roman, B. Lakshman Babu, “Failure management scheme for use in a flush air data systems”, Aircraft Design, 2001.
4. B. R. Cobleigh, Stephen A. Whitmore, Edward A Hoering, “Flush Airdata Sensor (FADS) systems calibration and procedures and result for blunt forebodies”, Dryden Flight Reserch Center, November 1999.
5. Camilla Di Fusco, “Ricostruzione degli angoli di incidenza e derapata del velivolo mediante elaborazioni dei dati aria basate su reti neurali”, Università degli studi di Pisa DIA, Tesi di laurea, 2006.
6. F. Schettini, “Studio delle problematiche di integrazione del sistema sensori nel Flight Control Systems di moderni velivoli ad elevate prestazioni”, Università degli studi di Pisa DIA, Dottorato di ricerca, 2005.
7. Howard Demuth, Mark Beale, “Neural Network Toolbox for use with Matlab”, Mathworks, September 2000.
8. Stefano Orsucci, “Sviluppo ed ottimizzazione delle logiche di gestione delle ridondanze in moderno Flight Control Systems Fly by Wire”, Università degli studi di Pisa DIA, Tesi di laurea, Aprile 2007.
9. A.Calia, R. Galatolo, V. Poggi, F. Schettini, ”Sviluppo di una sonda multipressione e di algoritmi di elaborazione per la ricostruzione dei parametri di volo”, XIX Congresso Nazionale AIDAA.
10. Giuseppe Del Nobile, “Manuale di Catia V5”, 2007
11. Fluent inc., “Gambit User’s Guide, February 2004

12. A.Quarteroni, L. Formaggia, “Mathematical modelling and numerical simulation of the cardiovascular system”, Gennaio 2002
13. Silvio Cammarata, “Reti Neuronali”, Etaslibri, 1990

Appendice A-Errori commessi sul Training set

In Appendice A sono riportate per alcune reti le immagini relative all'errore commesso sui punti del *training set* in funzione di α , β e velocità, ed in particolare per le reti a singola uscita in α e β con ingressi semplici addestrate con il metodo BR e per le reti ad ingressi semplici e doppia uscita addestrate con lo stesso metodo.

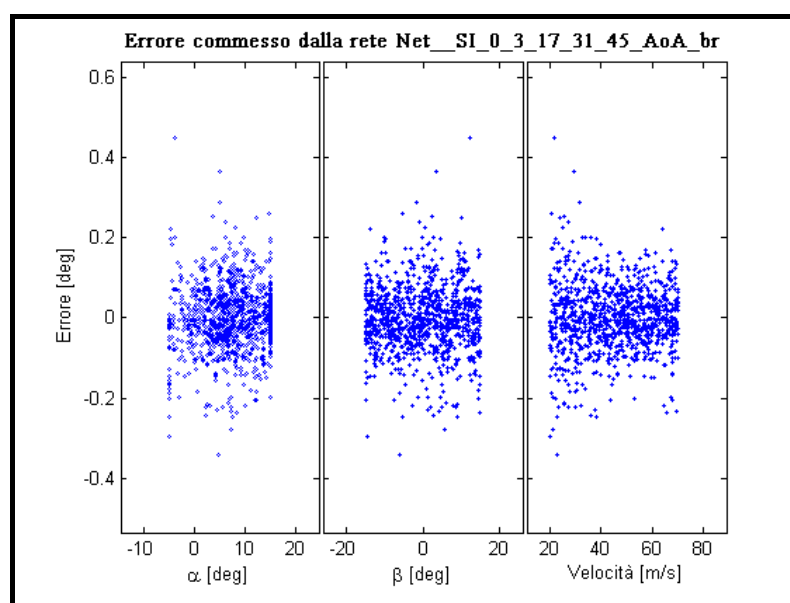


Fig. A.1 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoA_br sul *training set*

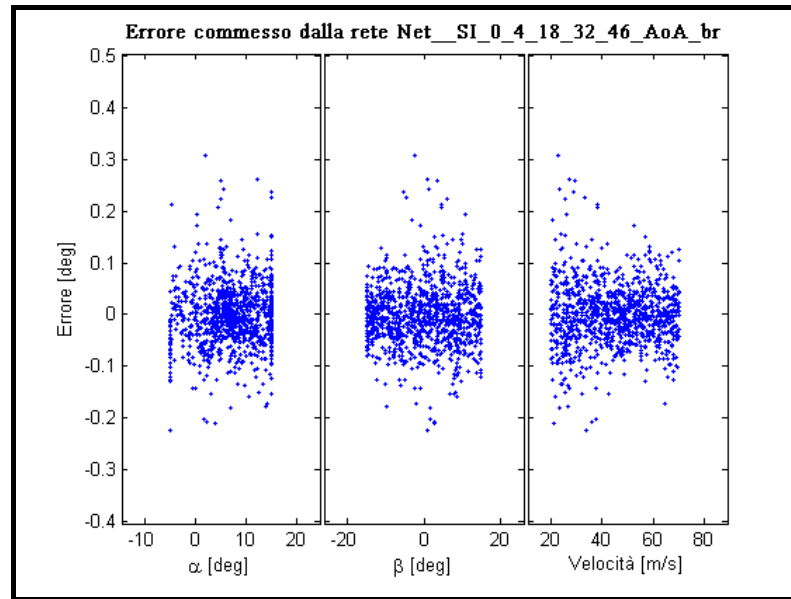


Fig. A.2 Errore commesso dalla rete *Net_NSI_0_4_18_32_46_Out_AoA_br* sul training set

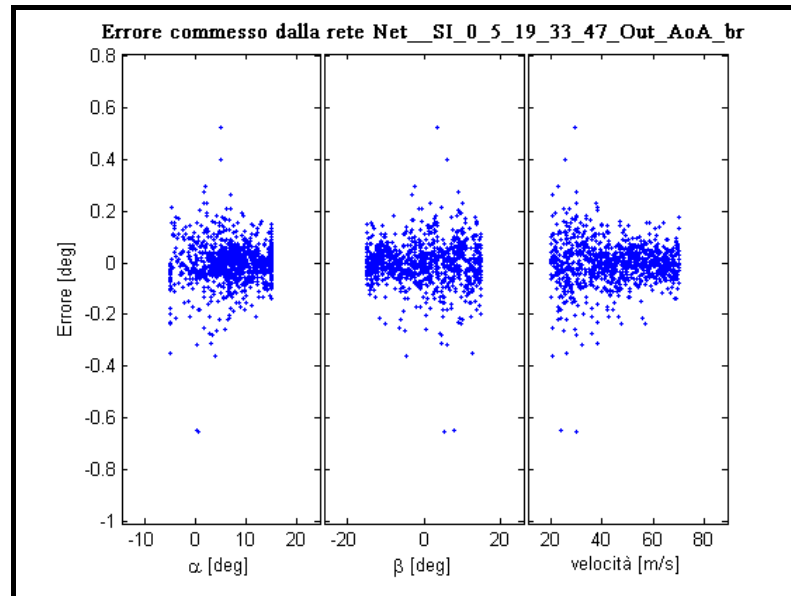


Fig. A.3 Errore commesso dalla rete *Net_NSI_0_5_19_33_47_Out_AoA_br* sul training set

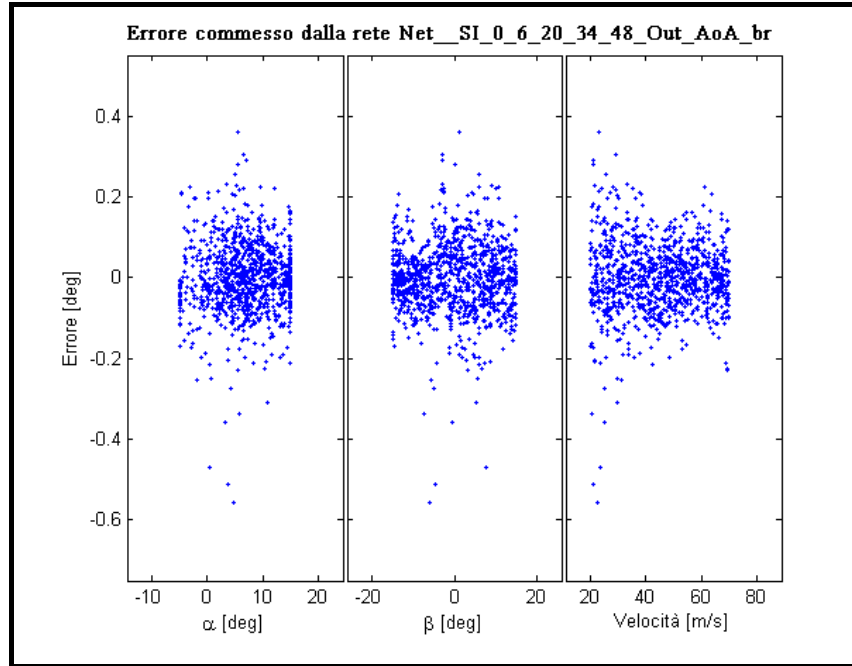


Fig. A.4 Errore commesso dalla rete Net_SI_0_6_20_34_48_Out_AoA_br sul training set

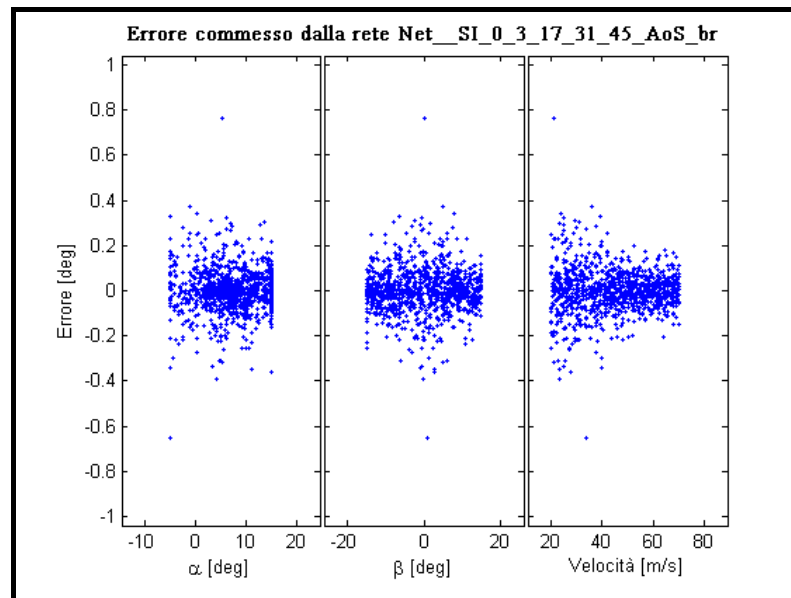


Fig. A.5 Errore commesso dalla rete Net_SI_0_3_17_31_45_Out_AoS_br sul training set

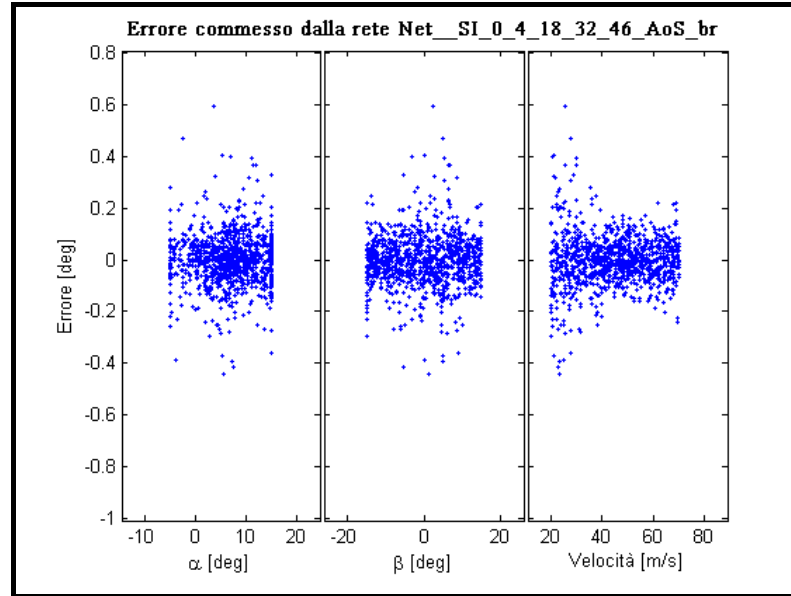


Fig. A.6 Errore commesso dalla rete *Net_NSI_0_4_18_32_46_Out_AoS_br* sul training set

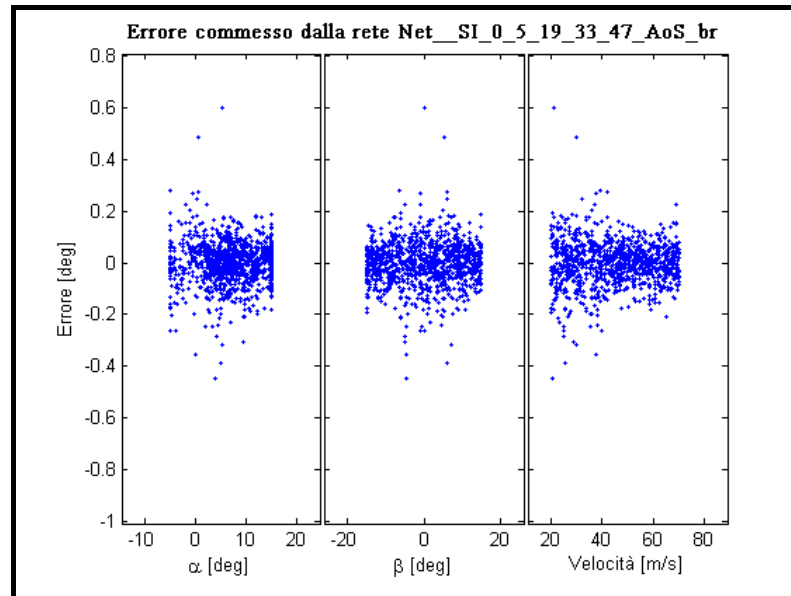


Fig. A.7 Errore commesso dalla rete *Net_NSI_0_5_19_33_47_Out_AoS_br* sul training set

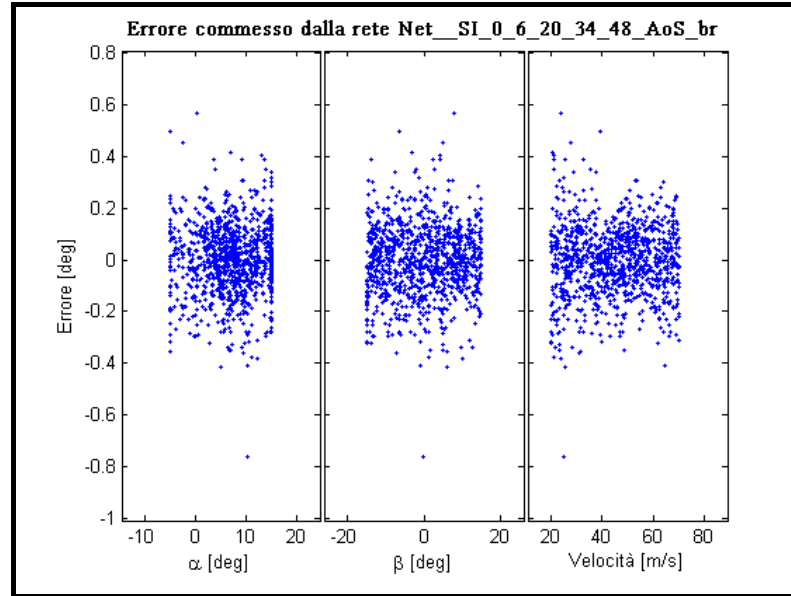


Fig. A.8 Errore commesso dalla rete Net_SI_0_6_20_34_48_Out_AoS_br sul training set

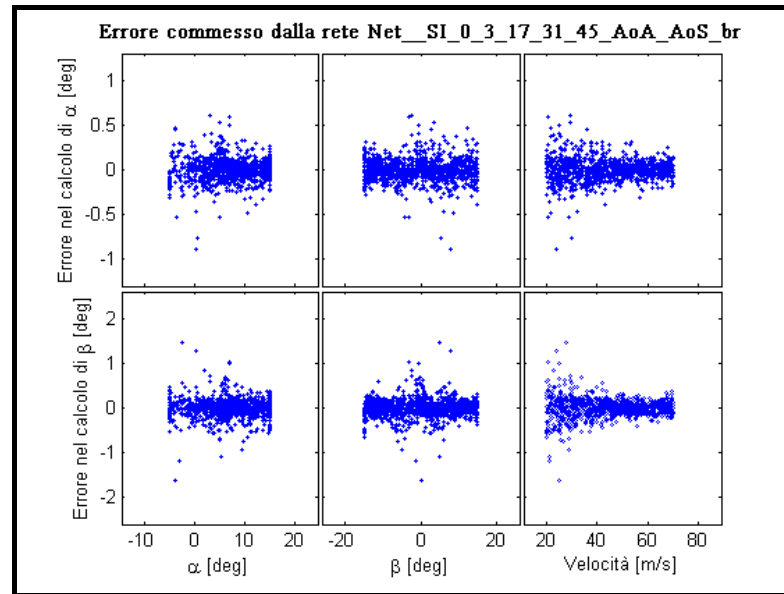


Fig. A.9 Errore commesso dalla rete Net_SI_0_3_17_31_45_Out_AoA_AoS_br sul training set

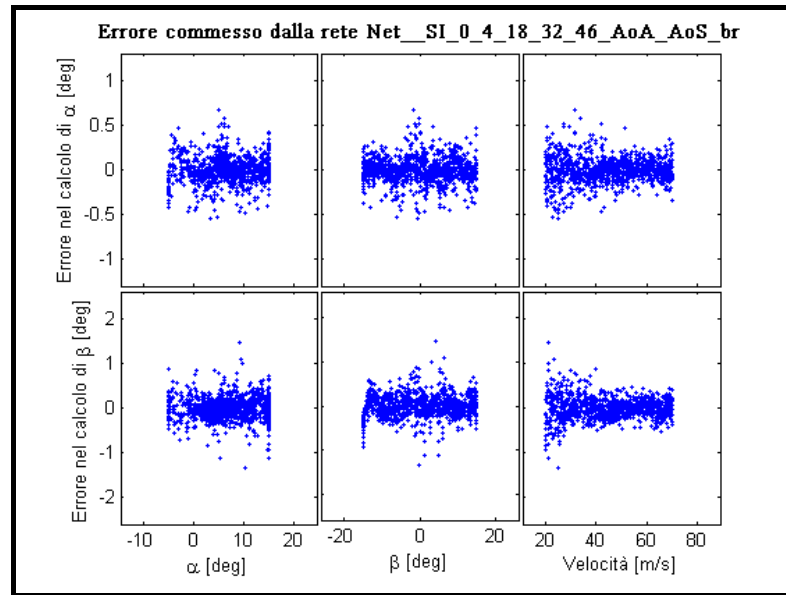


Fig. A.10 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoA_AoS_br
sul training set

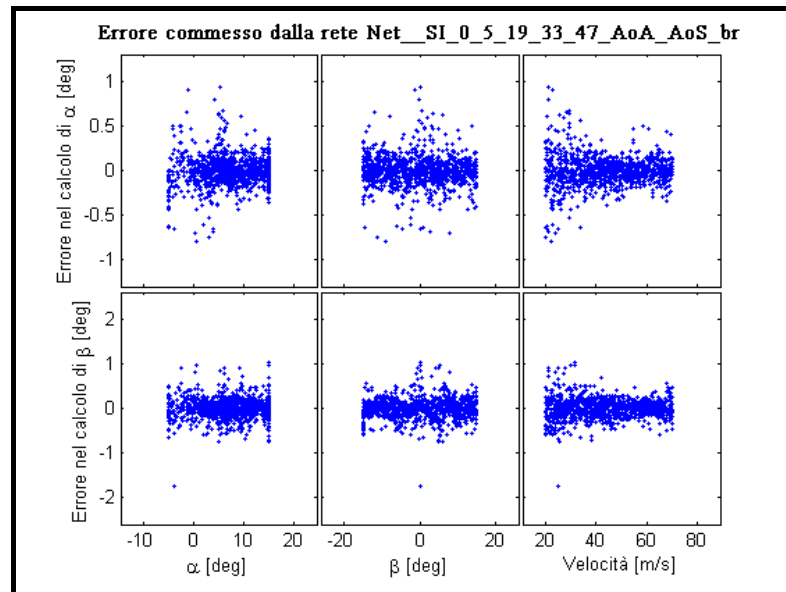


Fig. A.11 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoA_AoS_br
sul training set

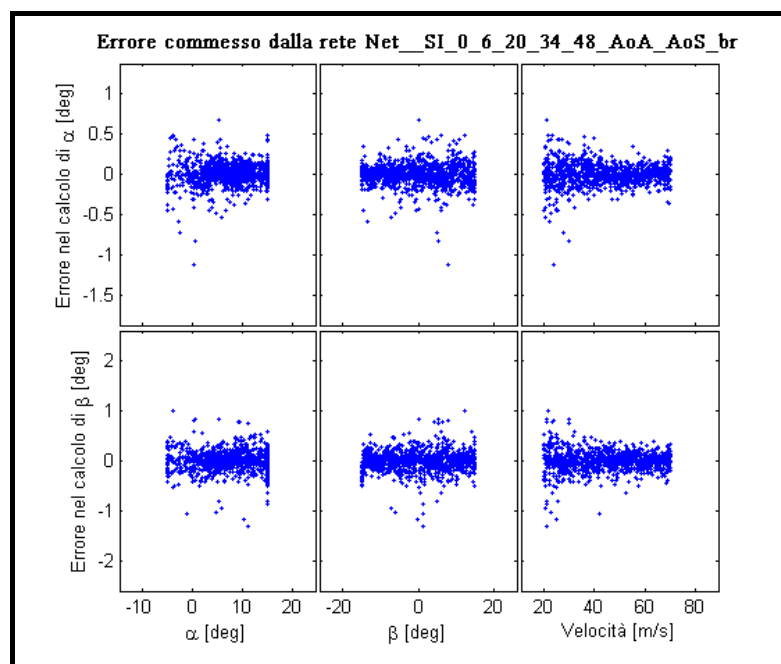


Fig. A.12 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoA_AoS_br
sul training set

Appendice B-Errori commessi sul testing set

In Appendice B sono riportati per le reti a singola uscita in α e β gli errori commessi sul testing set in funzione degli angoli di incidenza e derapata. Questi grafici riportano soltanto gli errori maggiori di 0.5 gradi per non appesantire le figure.

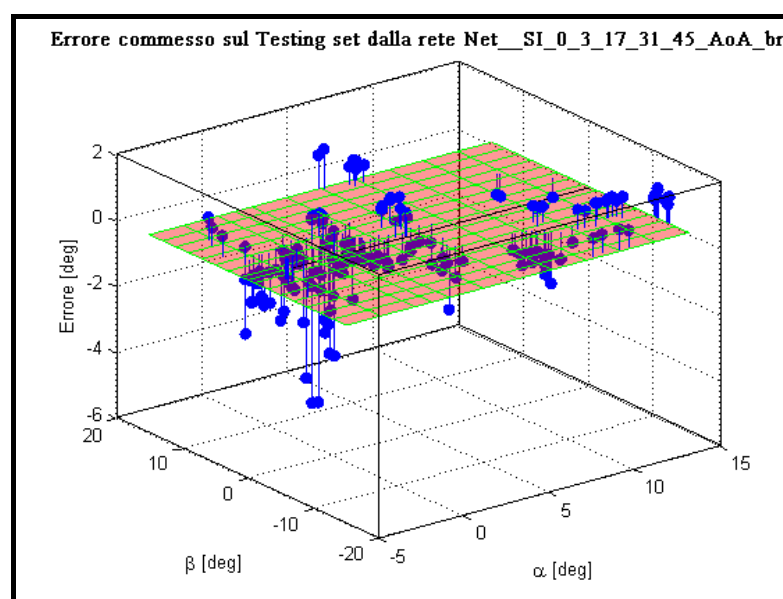


Fig.B.1 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoA_br sul testing set

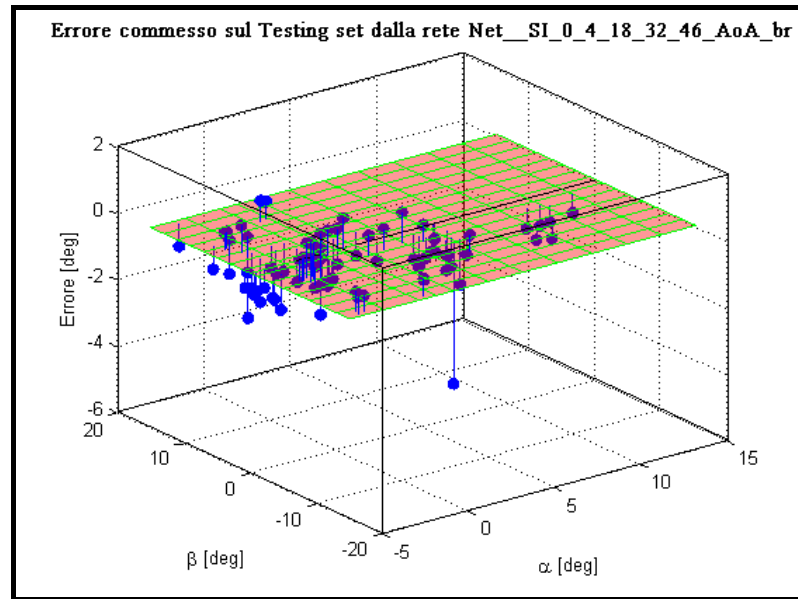


Fig.B.2 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoA_br sul testing set

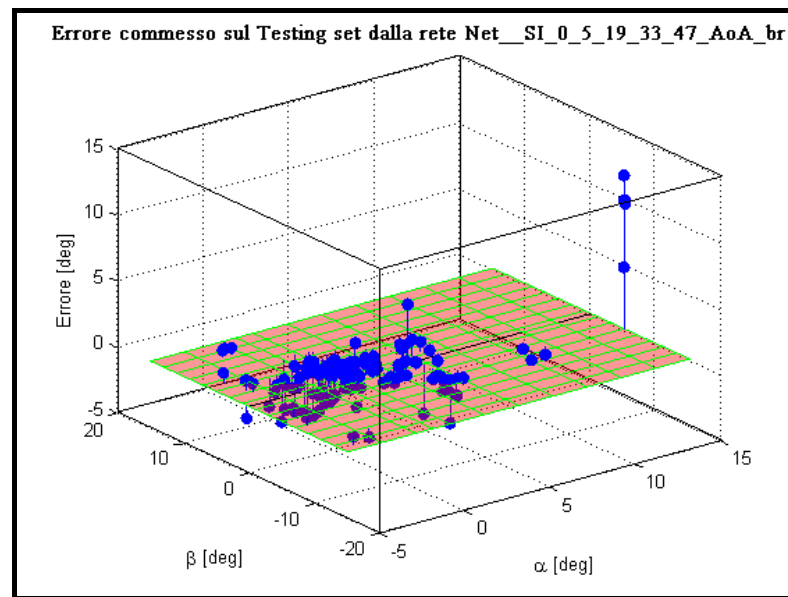


Fig.B.3 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoA_br sul testing set

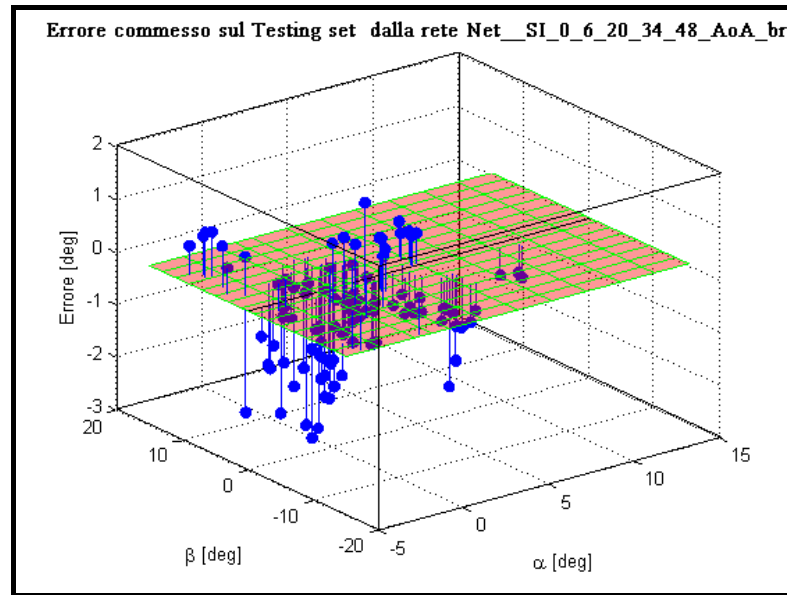


Fig.B.4 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoA_br sul testing set

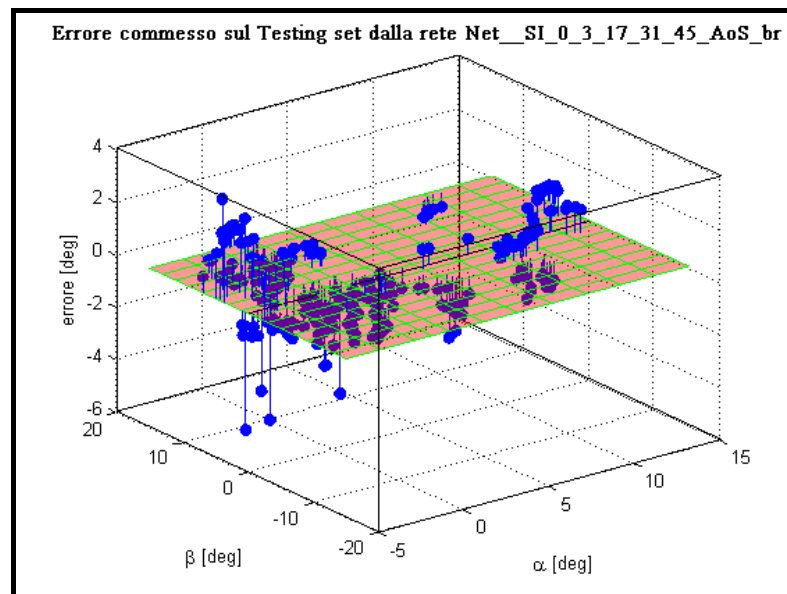


Fig.B.5 Errore commesso dalla rete Net_NSI_0_3_17_31_45_Out_AoS_br sul testing set

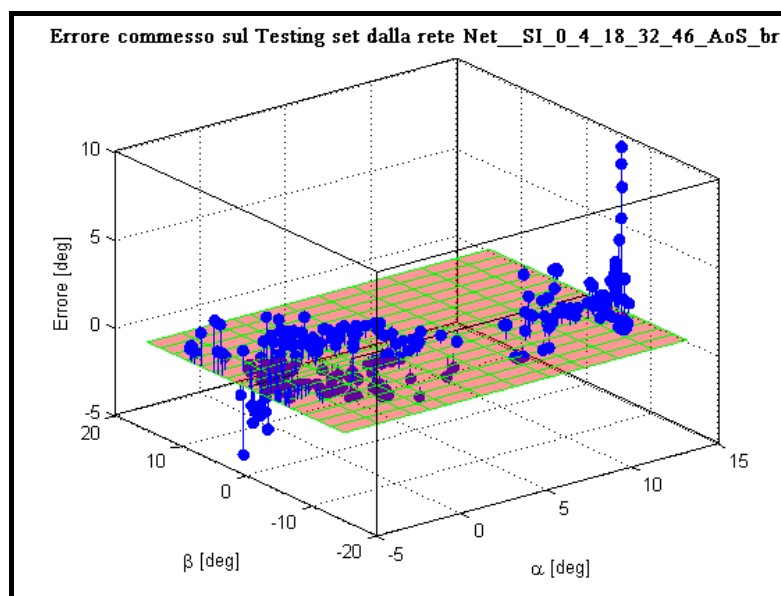


Fig.B.6 Errore commesso dalla rete Net_NSI_0_4_18_32_46_Out_AoS_br sul testing set

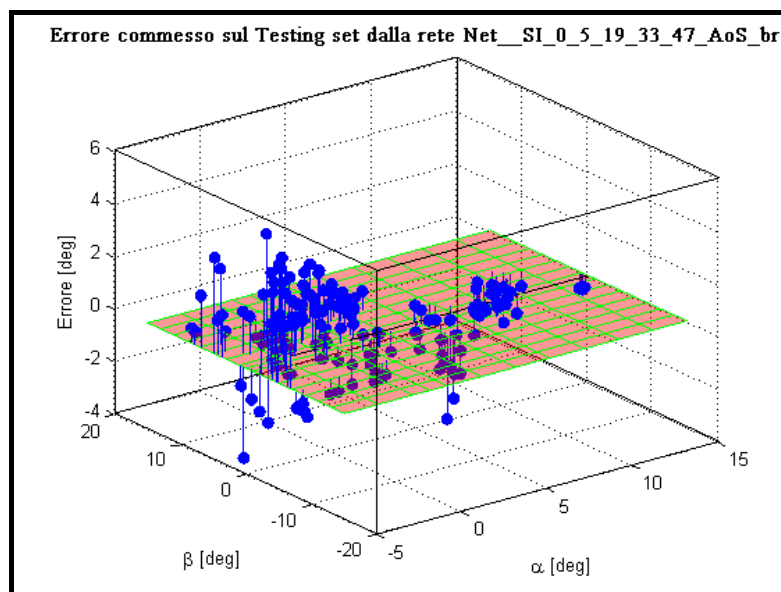


Fig.B.7 Errore commesso dalla rete Net_NSI_0_5_19_33_47_Out_AoS_br sul testing set

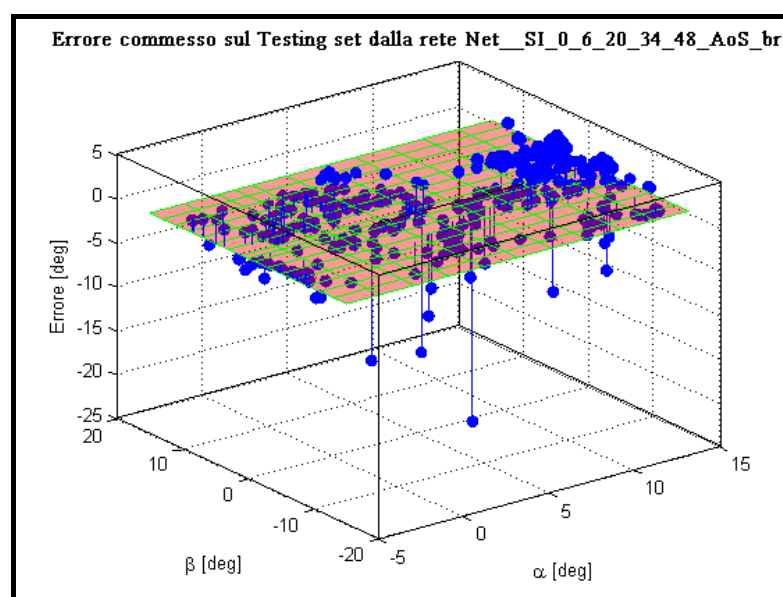


Fig.B.8 Errore commesso dalla rete Net_NSI_0_6_20_34_48_Out_AoS_br sul testing set

Ringraziamenti

La lista delle persone da ringraziare dopo questi lunghi anni di sofferenza è estremamente lunga e quindi cercherò di essere il più breve possibile. Inizio col ringraziare la mia ragazza Martina che è stata in grado di darmi quella forza e quella voglia che si stava piano piano esaurendo tra i libri, ed è riuscita a farmi concludere il più grande progetto (per ora) della mia vita. Ringrazio tutti i miei più cari amici e soprattutto Pierpaolo, compagno di tante avventure e di tante risate. Con lui abbiamo fatto tanti di quegli esami e preso tante di quelle bastonate che potremmo stare qui a raccontarle per mesi. Ringrazio Cosimo che soprattutto nell'ultimo periodo mi ha riempito di consigli (mai ascoltati) e prediche e che, grazie a me, dovrà sopportare una gastrite causata dall'abuso di caffè. Un ringraziamento doveroso va a tutti i ragazzi del calcetto, Bestia, Vairo, Pippo, Goal, Luchino, Gli Umalini Brothers, Marione, Capitan Becciu; il Baris, via i cosiddetti "ragazzi di Solano", che ho avuto il piacere di allenare ma che sono stati soprattutto degli amici dentro e fuori dal campo. Tra questi ringrazio soprattutto Andrea (Pipic) e Alessandro (Totò) con cui è nata da subito una amicizia particolare fatta di innumerevoli momenti passati insieme tra *Mac Donalds* e allenamenti e DLF, ecc. L'ultimo ringraziamento va a tutto il laboratorio di *Fly by Wire*, persone splendide che sono riuscite da subito a mettermi a mio agio e a creare un ambiente meraviglioso. Ringrazio, quindi, i miei relatori il Prof. Galatolo, il Prof. Denti, l'Ing. Schettini, l'Ing. Poggi, l'Ing. Calia e l'Ing. Di Rito che mi hanno aiutato a svolgere il lavoro della mia tesi nella maniera più accurata possibile, ma soprattutto ringrazio gli amici del Dipartimento: Francesco, Veronica, Alberto, GianPietro, Francesca, Valerio, Elena, Matteo con cui ci concedevamo delle lunghe pause caffè e dei momenti di relax per spezzare il lavoro. Mammà, Papà non mi sono dimenticato di voi. Vi ringrazio soprattutto per aver, nonostante le numerose lavate di capo e ramanzine (Papà sai di cosa parlo), sempre creduto in me, ma soprattutto per aver resistito alle mie scappatelle

lavorative, ai miei sbagli universitari e non, al mio pessimo carattere insomma (anche di questo sai di cosa parlo, vero Papà).

P.S. Alla faccia, meno male che dovevo essere breve.....

Pisa, 15 Luglio 2008-Giorno memorabile questo.